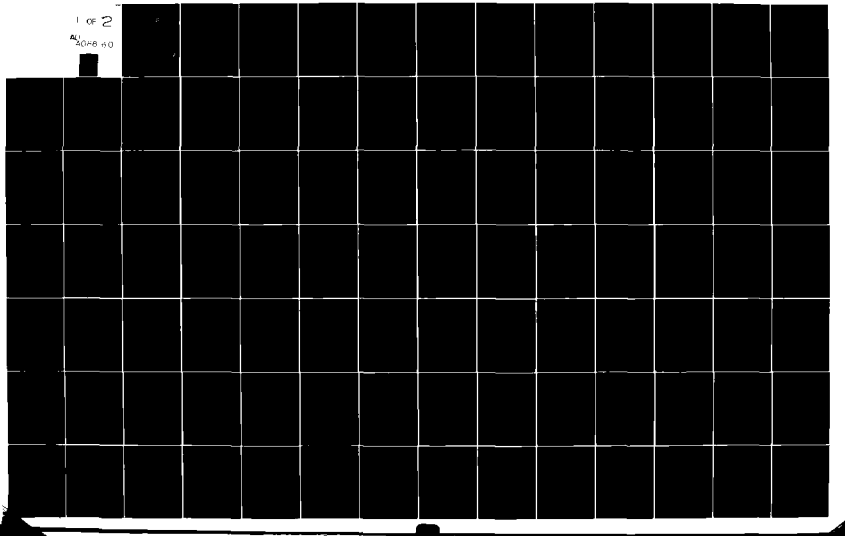
AD-A086 160 COMPUTER SCIENCES CORP FALLS CHURCH VA F/G 9/2
THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM. (QUICK). USE--ETC(U)
APR 80 D J SANDERS, P F MAYKRANTZ, J M HERRIN DCA100-78-C-0062
UNCLASSIFIED CCTC-CSM-UM-9-77-VOL-1-3 NL



DEFENSE COMMUNICATIONS AGENCY
COMMAND AND CONTROL TECHNICAL CENTER
WASHINGTON, D. C. 20301

IN REPLY REFER TO: C314

LEVEL

TO: RECIPIENTS

SUBJECT: Change 3 to Users Manual CSM UM 9-77, Volume I, Data Management Subsystem

1. Insert the enclosed change pages and destroy the replaced pages according to applicable security regulations.

2. A list of Effective Pages to verify the accuracy of this manual is enclosed. This list should be inserted before the title page.

3. When this change has been posted, make an entry in the Record of Changes.

FOR THE DIRECTOR:

11 Apr 81

112 Enclosures
Change 3 Pages

J. DOUGLAS POTTER
Assistant to the Director
for Administration

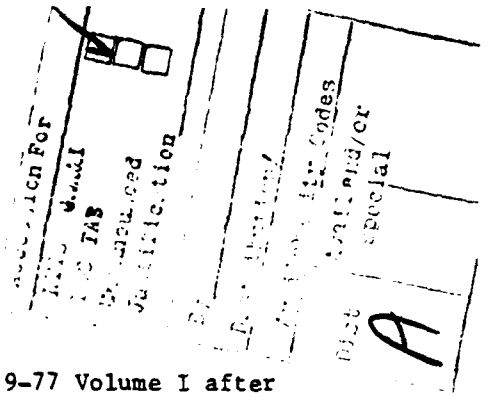
DCAN 78-C-0042

DTIC ELECTE JUN 24 1980 S D

DDC FILE COPY

This document has been approved for public release and sale; its distribution is unlimited.

80 6 23 049



EFFECTIVE PAGES - APRIL 1980

This list is used to verify the accuracy of CSM UM 9-77 Volume I after change 3 pages have been inserted. Original pages are indicated by the letter O, change 1 pages by the numeral 1, change 2 pages by the numeral 2, and change 3 pages by the numeral 3.

Page No.	Change No.	Page No.	Change No.
Title Page	0	44-44.2	3
ii	3	45-48	0
iii	0	49	3
iv	1	50	0
v	3	51	1
vi	1	52-52.2	3
vii	2	53	0
viii	3	54-54.2	3
ix	3	55-58	2
1	0	59	3
2	3	60	2
3-4	1	60.1-60.2	2
5-7	0	61-69	0
8	2	70-70.2	3
9-11	0	71-79	0
12	2	80-81	2
13-18	0	82	0
19	2	83	3
20-22	0	84	0
23	1	85-86.2	1
24-24.2	3	87-91	2
25-26	3	92	3
26.1-26.2	3	93	2
27-27.2	3	93.1-93.2	2
28	2	94	1
29	1	94.1-94.2	1
29.1-29.2	1	95	3
30-33	0	96-97	0
34-35	3	98-99	2
36	0	100-101.2	3
37-37.2	3	102	2
38	0	103-105.2	3
39-41	2	106.1	2
41.1	3	106.2-106.4	3
41.2	2	107	3
42	3	108-110	2
43	1		

Page No.	Change No.
111-111.2	3
112	2
113-114	1
115-116	2
116.1	2
116.2	1
117-121	0
122-124	2
125-126	1
126.1-126.4	3
127	0
128	1
129-130	3
130.1-130.5	3
131-132	0
133-134	2
135-135.2	3
136	0
137	2
138	0
139	2
140-141	1
142	0
143	3
144	1
145-146.2	3
147-147.2	3
148-152	3
153-154	0
155	3
156	0
157-158	3

ACKNOWLEDGMENT

This documentation was prepared under the direction of the Chief for Military Studies and Analysis, CCTC, in response to a requirement of the Studies, Analysis, and Gaming Agency, Organization of the Joint Chiefs of Staff. Technical support was provided by System Sciences, Incorporated under Contract Number DCA100-75-C-0019. Change set two was prepared under Contract Number DCA100-78-C-0035. Computer Sciences Corporation prepared change set three under Contract Number DCA 100-78-C-0042.

TABLES

Table

ABSTRACT

The computerized Quick-Reacting General War Gaming System (QUICK) will accept input data, automatically generate global strategic nuclear war plans, provide output summaries, and produce tapes to simulator subsystems external to QUICK. QUICK has been programmed in FORTRAN for use on the CCTC HIS 6080 computer system.

The QUICK Users Manual consists of four volumes: Volume I, Data Management Subsystem; Volume II, Weapon/Target Identification Subsystem; Volume III, Weapon Allocation Subsystem, Volume IV, Sortie Generation Subsystem. The Users Manual complements the other QUICK Computer System Manuals to facilitate application of the war gaming system. This volume, Volume I, provides instructions for using the Data Management Subsystem. It is intended for the CCTC user/analyst who is concerned with preparing the data base for a war game, selecting optional features of QUICK, designating control parameters, submitting computer jobs, and analyzing computer output. Companion documents are:

a. PROGRAM MAINTENANCE MANUAL
   Computer System Manual CSM MM 9-77, Volume I
   Computer System Manual CSM MM 9-77, Volume II
   Computer System Manual CSM MM 9-77, Volume III
   Computer System Manual CSM MM 9-77, Volume IV
   Provides detailed instructions for maintenance of the system.

b. TECHNICAL MEMORANDUM
   Technical Memorandum TM 153-77
   Provides a nontechnical description of the system for senior management personnel.

SECTION 1. GENERAL

1.1 Purpose

This manual is intended to inform the CCTC user/analyst on how to prepare control cards; structure execution (run) decks; prepare computer job requests; and understand the associated computer output, to include the recognition of error messages for the Data Management subsystem of QUICK. It complements information contained in the Program Maintenance Manuals on the QUICK system. The abstract of this document references other documents describing QUICK.

1.2 Program Environment - QUICK System Overview

The QUICK-Reacting General War Gaming System (QUICK) is a unique analytical tool which provides a comprehensiveness to strategic war gaming that has not been available through other computerized models. QUICK is designed to assist in the study of strategic conflicts involving a large-scale exchange of nuclear weapons. Toward this end, the system encompasses three major capabilities which are applicable to a wide range of studies: first, for a given offensive missile and bomber force and a specific set of targets, QUICK produces a detailed plan of attack which is near optimum for the conditions specified by the user. Second, it provides an expected-value estimate of the results of that attack. Finally, it produces input tapes to simulator subsystems external to QUICK.

QUICK is structured into four major subsystems: Data Management, Weapon/Target Identification, Weapon Allocation, and Sortie Generation. The principal tasks associated with each of these functional subsystems are summarized below.

a. Data Management: Assembles and reformats the target and non-target data required for a particular plan

b. Weapon/Target Identification: Selects and processes the Red and/or Blue Forces prespecified for a particular plan

c. Weapon Allocation: Allocates offensive weapons to selected targets

d. Sortie Generation: Prepares and evaluates missile and bomber attack plans.

Figure 1 displays the modules which comprise each subsystem.

Figure 2 illustrates the communication with the Central Operations Processor (COP) or executive software and the entire procedural and informational flow within the QUICK system. The communication lines infer

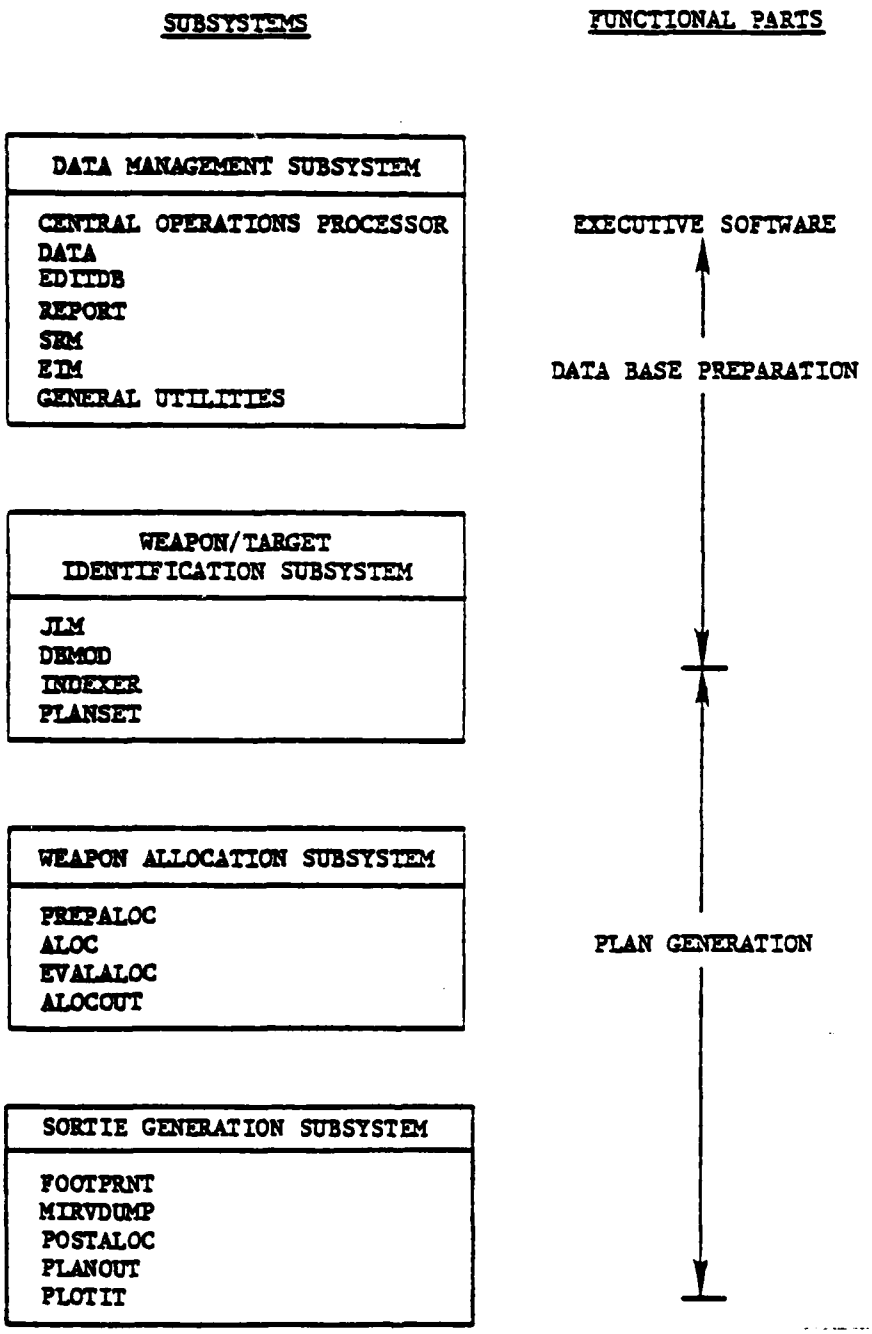


Figure 1. Major Subsystems of the QUICK System

b. Phrase Types

- o Relational Phrases - Sublevels are used to define comparative or logical meaning according to operators
- o Restricted Relational Phrases - Used to set attributes and/or variables
- o Elements - Used to set single elements such as value elements, special words or display names.

The clause types define the overall syntax of the clause, and the phrase types define which phrases fall into or are connected by the overall syntax.

3.3 QUICKs Dictionary

All permissible words within the developed language that have meaning to the COP are contained within the 'Dictionary' (a list of tables stored in the data base). These words contain attributes (as employed for target and weapon definition), plus other words necessary for the syntax. Words within the dictionary are grouped as outlined in previous subsections. Based on these groupings, tables 6 through 11 present the entire list of words defined within QUICKs command language plus comments on how each word may be used. The list of attributes are defined in appendix A of this manual.

Occasionally a user will wish to enter an alphanumeric string which ordinarily will be recognized by the dictionary but which the user wishes to treat as an alphanumeric constant. The user may do so by enclosing the string in quotation marks. For example, if the user wishes the string AS to be treated as alphanumeric rather than a null, he should enter it thus - "AS".

A sentence command written with entries not contained within the cited tables are words foreign to the language and may cause computer execution aborts. In addition, only certain combinations of words from the "dictionary" (such as verbs, adverbs, etc.) have structural meaning. The entire QUICK system generates its final output through a series of selections of individual program modules as defined by the verb. The selected module then can act (or interpret) only on those sentence patterns that request outputs produced within the program. This is also analogous to formal English where individual thoughts are expressed in separate paragraph constructions.

Many of the Adverbs in table 7 introduce clauses which are used by a number of verbs. These clauses are described in the following paragraphs.

3.3.1 DEFINE Clause. A DEFINE clause consists of a single equals phrase in which the subject is used as an alphabetic variable name. (The subject should not be an attribute.) The object of the phrase should be a mathematical formula combining attributes and numeric values plus any alphabetic string which the user intends to employ as the variable name of this or any other DEFINE.

Table 6. QUICK's Text English Verbs
(Parc 1 of 2)

VERB	MODULE	DESCRIPTION
ALLOCATE	ALOC	Executes the ALOC module
ALTER	REPORT	Makes updates and changes to a previously designed display
ASSIGN	JLM	Builds the Assignment Table
ASTERISK	JLM	Makes Damage Assessment Tape from a JAD format tape and the Data Base
BUILD	EIM	Instructs the External Interface Module to build a file
CHANGE	DATA	Updates data element(s)
CREATE	DATA	Creates new data elements
DATAMAKE	DATAMAKE	Executes the DATAMAKE module
DELETE	DATA	Deletes records
DESIGN	REPORT	Constructs and saves a REPORT module display
DGZSELECT	ALOCOUT	Executes the ALOCOUT module
EDIT	EDITDB	Executes the EDITDB module
EVALUATE	EVALALOC	Executes the EVALALOC module
FOOTPRNT	FOOTPRNT	Executes the FOOTPRNT module
FRAMIS	FRAMIS	Executes the FRAMIS module
INDEX	INDEXER	Executes the INDEXER module
MODIFY	DBMOD	Executes the DBMOD module
MIRVDUMP	MDUMP	Executes the MDUMP module
PLANOUT	PLANOUT	Executes the PLANOUT module

Table 6. QUICK's Text English Verbs
(Part 2 of 2)

VERB	MODULE	DESCRIPTION
PLANSET	PLANSET	Executes the PLANSET module
PLOTDATA	EIM	Creates geography plot tapes
PLOTIT	EIM	Creates sortie plot tapes
POSTALOC	POSTALOC	Executes the POSTALOC module
PREPARE	PREPALOC	Executes the PREPALOC module
PRINT	REPORT	Prints some user defined (built by DESIGN) display
RESTORE	SRM	Brings the contents of an IDS data base from magnetic tape to a disk file
SAVE	SRM	Stores the contents of an IDS data base on a magnetic tape
SELECT	JLM	Selects records from a JAD file

Table 7. QUICK's Adverb List
(Part 1 of 4)

ADVERB	DESCRIPTION
ABTAPE	Requests PLANOUT to produce an AB tape
ACARD	Describes a non-MIRV missile sortie to PLANOUT
ALPHAS	Defines the CLASS, TYPE, DESIG, and category relationships for the Assignment Table
ATTACKERS	Defines the attacking weapon systems
CCARD	Describes sortie change to PLANOUT
COUNTRIES	Introduces countries to be processed by EVALALOC
DEFENDERS	Defines the exemplar targets and associated values
DEFINE	Describes a user defined variable for use in either printed or tape/file output
DISPLAY	Names a REPORT module display
EQUATE	Describes a footprint equation
FIELDS	Lists attributes in the data base to be tested to see if they are within defined ranges
FILE	Used to define type of file EIM should build
FINDMIN	Gives number of iterations for FINDMIN subroutines of ALOCOUT
FIX	Defines weapon fixed assignments
FIXOFF	Defines weapon fixed assignments with latitude/longitude offsets
FLAGREST	Describes allocator flag restrictions
FORMAT	Describes the format to be used in the creation of printed or tape/file output

Table 7. (Part 2 of 4)

ADVERB	DESCRIPTION
FUNCOM	Introduces the function and commands for PLANOUT
GAMETIME	Introduces the game time for PLANOUT
ICARD	Describes insert in sortie to PLANOUT
IF	Defines limits for equations in FOOTPRNT
KEEPING	Lists the DESIGs which are not to be deleted from the data base after ASTERISK is run
LOCREST	Describes allocator country location restrictions
MINRANGE	Describes modifications to the RNGMIN attributes
MIRVREST	Describes allocator MIRV weapon system restrictions
MISTME	Describes time coordinated missile attacks
MODRANGE	Describes modifications to the RANGE and RANGEREF attributes
MSLCOR	Describes missile launch timing
ONPRINTS	Defines user controlled printed output
OMITTING	Keeps duplicate targets from being added to the data base
ORDER	Allows the user to input the order that the classes will be entered into the data base.
PLANE	Introduces plane-type indexes for PLANOUT
PLAYERS	Describes valid country codes and the regions that they are in
PRIORITY	Introduces a complexing priority scheme
PUNCH	Produces cards containing final allocator multipliers

Table 7. (Part 3 of 4)

ADVERB	DESCRIPTION
READMUL	Reads in pre-set allocator multipliers
RECALC	Indicates recalculation mode for various modules
REEQUATE	Describes changes to a previously entered footprint equation
REPLACING	Causes duplicate targets to be replaced in the data base.
SAME	Describes a record similar to the one being created which is to be used for default values.
SETTING	Introduces a clause that stores data elements
SMAT	Describes values and updates for the SMAT array of the allocator.
SORT	Describes the sort order for printed or tape/file output
STRIKE	Request PLANOUT to produce a STRIKE tape
SUPRESSING	Suppresses data value editing during data creation
TGTMOD	Introduces target value changes for EVALALOC
UICLASSES	Describes CLASS names of target classes to be treated as "U/I" by DBMOD
UNIT	Gives tape/file logical unit number
USING	Requests data value editing during data creation
VNOPTION	Used to select a complexing option
WHERE	Describes subset of data base on which verb's action is to be performed
WITH	Describes the relationships that must be met between selected attributes in the data base.

Table 7. (Part 4 of 4)

ADVERB	DESCRIPTION
WPNMOD	Introduces weapon characteristic changes for EVALALOC; indicates calculation of weapon overallocation data and salvo information for PREPALOC.

Table 8. QUICK's Text English Adverbs Usage
(Part 1 of 3)

ADVERBS	CLAUSE TYPE	PHRASE TYPE	VERB(S)
ABTAPE	Null	Null	PLANOUT
ACARD	Sequence	Elements	PLANOUT
ALPHAS	Sequence	Elements	ASSIGN
ATTACKERS	Sequence	Elements	PLANSET
CCARD	Sequence	Elements	PLANOUT
COUNTRIES	Sequence	Elements	EVALUATE
DEFENDERS	Single	Restricted Relational	PLANSET, DATAMAKE
DEFINE	Single	Restricted Relational	DESIGN, ALTER, BUILD
DISPLAY	Sequence	Elements	DESIGN, ALTER, PRINT
EQUATE	Sequence	Elements	FOOTPRNT
FIELDS	Sequence	Elements	EDIT
FILE	Single	Elements	BUILD
FINDMIN	Single	Element	DGZSELECT
FIX	Sequence	Elements	PREPARE
FIXOFF	Sequence	Elements	PREPARE
FLAGREST	Sequence	Elements	ALLOCATE
FORMAT	Sequence	Elements	DESIGN, ALTER, BUILD
FUNCOM	Sequence	Elements	PLANOUT
GAMETIME	Sequence	Elements	PLANOUT
ICARD	Sequence	Elements	PLANOUT

Table 8. (Part 2 of 3)

ADVERB	CLAUSE TYPE	PHRASE TYPE	VERB(S)
IF	Boolean	Relational	FOOTPRNT, PLANOUT
KEEPING	Sequence	Elements	ASTERISK
LOCREST	Sequence	Elements	ALLOCATE
MINRANGE	Sequence	Elements	ALLOCATE
MIRVREST	Sequence	Elements	ALLOCATE
MISTME	Sequence	Elements	PLANOUT
MODRANGE	Sequence	Elements	ALLOCATE
MSLCOR	Sequence	Elements	PLANOUT
ONPRINTS	Sequence	Elements	(ALL VERBS)
OMITTING	Null	Null	SELECT
ORDER	Sequence	Elements	SELECT
PLANE	Sequence	Elements	PLANOUT
PLAYERS	Sequence	Elements	ASSIGN
PRIORITY	Sequence	Elements	PLANSET, DATAMAKE
PUNCH	Sequence	Elements	ALLOCATE
READMUL	Sequence	Elements	ALLOCATE
RECALC	Null	Null	ALLOCATE, PLANOUT, INDEX, PREPARE
REEQUATE	Sequence	Elements	FOOTPRNT
REPLACING	Null	Null	SELECT
SAME	Sequence	Elements	CREATE

Table 8. (Part 3 of 3)

ADVERB	CLAUSE TYPE	PHRASE TYPE	VERB(S)
SETTING	Sequence	Restricted Relational	PLANSET, MODIFY, DESIGN PREPARE, CREATE, ALTER, CHANGE, PLOTDATA, SELECT, ALLOCATE, EVALUATE, POSTALOC, PLOTIT, PLANOUT, DATAMAKE
SMAT	Sequence	Elements	ALLOCATE
SORT	Sequence	Elements	DESIGN, ALTER, BUILD
STRIKE	Null	Null	PLANOUT
SUPRESSING	Null	Null	CREATE
TGTMOD	Sequence	Elements	EVALUATE
UICLASSES	Sequence	Elements	MODIFY
UNIT	Single	Element	SELECT, SAVE, RESTORE BUILD
USING	Null	Null	CREATE
VNOPTION	Null	Null	INDEX, DATAMAKE
WHERE	Boolean	Relational	DESIGN, PRINT, ALTER, CHANGE, DELETE, SELECT, BUILD, EDIT
WITH	Sequence	Relational	INDEX, EDIT, DATAMAKE
WPNMOD	Sequence	Elements	EVALUATE, PREPARE

Table 9. QUICKs Text English Special Words
(Part 1 of 3)

SPECIAL WORD	USE	DESCRIPTION
A	SORT adverb PLANOUT verb	Same as ASCENDING for SORT, Implies air burst for PLANOUT
ACOS	EQUATE, REEQUATE adverbs	Arc-cosine
ACOT	EQUATE, REEQUATE adverbs	Arc-cotangent
AFTER	ALTER verb* FORMAT adverb	Introduces additions to format after indicated PAGE, LINE, etc.
ASCENDING	SORT adverb	Lowest values will be first
ASIN	EQUATE, REEQUATE adverbs	Arc-sine
ATAN	EQUATE, REEQUATE adverbs	Arc-tangent
C	PLANOUT verb	Value for "CALOFF" field, implies DLATOF and DLONGOF represent actual DGZ
COS	EQUATE, REEQUATE adverbs	Cosine
COT	EQUATE, REEQUATE adverbs	Cotangent
D	SORT adverb	Same as DESCENDING
DESCENDING	SORT adverb	Highest values will be first
EXCLUDE	FLAGREST, LOCREST, and COUNTRIES adverb	Indicates following items are excluded from consideration
EXP	EQUATE, REEQUATE adverbs	Exponential

This DEFINE is calculated only once but it included a DEFINE that is calculated for each record processed.

3.3.2 FORMAT Clause. The FORMAT clause is used by both the REPORT and EIM modules to define the format of output tapes, files and printed reports. The FORMAT clause consists of a series of subclauses introduced by one of the four special words: PAGE, LINE, HEADER, and TRAILER.

3.3.2.1 PAGE. This special word begins a set of headers, trailers and lines. Each set is produced separately. The PAGE special word has no elements following it. A FORMAT clause need not be started by a PAGE special word; one will be assumed if the user omits it.

3.3.2.2 HEADER. This special word introduces a string of elements that define one or more lines which are to appear at the top of every physical page. The headers will be produced in the order input.

3.3.2.3 TRAILER. This special word introduces a string of elements that define one or more lines which are to appear at the bottom of every physical page. Trailers will be produced in the order input.

3.3.2.4 LINE. This special word introduces a string of elements that make up the body of the report or file. Lines are of two types: those that contain attributes and DEFINEs that are not sums or products and those which contain no attributes and only DEFINEs which are sums or products. The former will produce one line for every logical record retrieved from the data base; the latter will produce only one line.

3.3.2.5 Special Words and Elements. HEADER, TRAILER and LINE special words are followed by a series of elements or element sets which may be any of the following:

- Alphabetic - used as a constant as input
- Attribute - an attribute will be displayed either in a format calculated by the module or provided by the user via an IN phrase
- Define variable name - a variable will be displayed either in a format calculated by the module or provided by the user via an IN phrase
- IN special word - this special word must follow an attribute or variable name and must be followed by a FORTRAN format (viz. F10.6). In this way the user specifies the format in which the attribute or variable is to be displayed
- Long String - an alphabetic string enclosed by either of the characters (' or ") appears in the format as input
- Numeric - followed by the special word SPACES (or X) produces as many blank spaces as is specified by the numeric.

3.3.3 ONPRINTS Clause. This is the general clause used to select the optional print of any verb.

3.3.4 SETTING Clause. This clause is used to specify values for attributes. It may contain either equals or like phrases. Many modules will restrict the attributes which may be used. The user should be aware that the use of an extended equals (e.g., (LAT,LONG)=(501020N,1200000W) &(481020N,1152501W)) has special meaning within the context of the CHANGE verb.

3.3.5 SORT Clause. This clause is used to specify the sort order to be used in the case of an output file or report. The clause consists of a series of pairs of elements. The first element of the pair is either an attribute or a defined variable name. The second element of the pair is one of the two special words: ASCENDING (or A) or DESCENDING (or D). The order of the sort is the order of the element pairs, the first pair specifying the major sort. An example of a sort clause is:

SORT LAT ASCENDING LONG DESCENDING

3.3.6 UNIT Clause. This clause is used to specify a logical tape or file unit number. It has the form:

UNIT logical unit number

3.3.7 WHERE Clause. This clause is used to specify a subset of the data base for processing by the module called. It has the form:

WHERE a_1 [AND a_2 AND a_3 AND . . . AND a_n]

or

WHERE a_1 [OR a_2 OR a_3 OR . . . OR a_n]

where the a_i are logical expressions. The simplest logical expression is a relational phrase. However, a logical expression can also be of the form: NOT a_i

where a_i is a logical expression. This form implies the negation of the expression a_i. Finally, a logical expression may be of the form:

WHERE (a_1 [AND a_2 AND . . . AND a_n])

or

WHERE (a_1 [OR a_2 OR . . . OR a_n])

where the a_i are logical expressions

The user should be aware that the use of collections and/or extended EQUALS (i.e., (LAT, LONG) = (501020N,1200000W) & (481020N,1152501W)) creates internally a complex logical expression and, if imbedded within another complex logical expression, should be enclosed in parentheses.

SECTION 4. CENTRAL OPERATIONS PROCESSOR

4.1 General Purpose

The Central Operations Processor (COP), or QUICKs executive software module, acts as an intermediary between the user and the operating system, and provides the communication between the QUICK system and the integrated data base. The COP acts accordingly by interpreting user commands (explained in section 2 of this manual) written in imperative text English formats.

Inputs to the COP are completely generalized to the degree that the text English commands are. The COP itself generates very little output which is one of the main functions of each module that the COP executes. Standard inputs and outputs are discussed within the following subsections.

4.1.1 BOOT Entry. Given an integrated data base, the COP operates free form. However, the data base must be initially created as to the structure and specifications of data items within the structure. Since the COP must operate with the integrated data base, a special entry (or mode of execution) is necessary to inform the COP that data follows whereby the data base itself will be defined. This mode of execution is performed within the BOOT module which is a special portion of the COP. The BOOT module will read data and create the data base structure. This creation is a one-time operation and once successfully executed, only minor updates will be necessary. Separate subsections, following the discussion on the COP, will outline the BOOT module operation.

4.2 Inputs to the COP

There is not standard input to the COP, only the previously cited generalized text English commands. From the verb within each command, the COP directs exeuction of each related module. Each module within this, and other user manuals will further detail the nature of the inputs.

Appendix E of this manual presents the Job Control Language (JCL) necessary to execute the COP for batch submittals as well as a series of commands for timesharing submittals.

4.3 COPs Output

The standard output of the COP is a print of the input commands preceded by an introductory header (figure 3). If the user input consists of a CREATE, CHANGE, or DELETE verb, it will be accompanied by a sequence number which will enable the user to cross-reference this type of input

with the data module quality control list. Further, the standard and/or optional output of each module is preceded by a banner print of the module's overlay link name (figure 4). This banner appears each time a new module is called and will only appear before the first instance of consecutive calls to the same module.

The COP error messages appear in figure 5.

```
(1) ***** CENTRAL OPERATIONS PROCESSOR *****

(2) BUILD FILE OTHER UNIT 42 SORT LAT A
        DEFINE COUNT=COUNT+1
        DEFINE VALTOT=VALTOT+VALUE
        WHERE SIDE=BLUE & CLASS=MISSILE
        FORMAT LINE VALTOT IN F15.2 COUNT IN I13
               LINE CLASS IN A6 TYPE IN A6
                    LAT IN F6.2 LONG IN F7.2
               LINE DESIG IN A5 LAT IN F6.2
                    LONG IN F7.2 VALUE IN F7.2
        ONPRINTS
```

HEADING	MEANING
(1)	COP input print header
(2)	User input

Figure 3. Central Operations Processor, Input Print

```
RRRRRR      EEEEEEEE    PPPPPP        0000      RRRRRR      TTTTTTTT
RRRRRRR     EEEEEEEE    PPPPPPP     000000      RRRRRRR     TTTTTTTT
RR   RRR    EE          PP   PPP   000  000    RR   RRR       TT
RR    RR    EE          PP    PP   00    00    RR    RR       TT
RR    RR    EE          PP    PP   00    00    RR    RR       TT
RR   RRR    EE          PP   PPP   00    00    RR   RRR       TT
RRRRRRR     EEEEEE      PPPPPPP    00    00    RRRRRRR        TT
RRRRRR      EEEEEE      PPPPPP     00    00    RRRRRR         TT
RR  RRR     EE          PP         00    00    RR  RRR        TT
RR   RRR    EE          PP         00    00    RR   RRR       TT
RR    RR    EE          PP         00    00    RR    RR       TT
RR    RR    EE          PP         000  000    RR    RR       TT
RR    RR    EEEEEEEE    PP           000000     RR    RR       TT
RR    RR    EEEEEEEE    PP             0000     RR    RR       TT
```

Figure 4. COP Banner Print

18 SYNTAX ERROR - ILLEGAL ITEM IN VALUE EXPRESSION

Value expression contains illegal item.

19 SYNTAX ERROR - ILLEGAL ELEMENT

Item included in clause of elemental adverb which is illegal according to syntax directory.

20 SYNTAX ERROR - TOO MANY PHRASES

Adverb defined as "single" has more than one phrase.

21 SYNTAX ERROR - RELATION NOT RIGHT IN SEQUENCE

Normal sequence of relational phrase violated.

22 TYPE (I5) VALUE (I5) (A12)

Arguments are the type, value and alphabetic representation of the offending item. This message appears following most of the messages that begin with "SYNTAX ERROR"

The values for type are as follows:

1 - operator	5 - special word
2 - long string delimiter	6 - attribute
3 - verb	9 - alphanumeric value
4 - adverb	10 - numeric value

If type is 1, 3, 4, 5, or 6 - value will contain the numeric identifier associated (see Appendixes C and D).

If type 1 or 2, no alphabetic representation appears.

23 INPUT TABLES EXCEEDED, TYPE (I2)

Tables built by subroutine ERRFND have been exceeded. Argument indicates type of table in which error occurred.

The following are the table types:

1 - Numeric entries
2 - Attributes entered
3 - Alphabetic value entries
4 - Items entered (one entry for each syntactically distinct portion of the input)

Figure 5. (Part 3 of 4)

24 LONG STRING TOO LONG

Input long string exceeds 120 characters.

25 INPTRN - UNBALANCED COLLECTION

Number of items in object collection(s) does not agree with number in subject.

26 INPTRN - BOOLEAN STATEMENT WRONG

Error in boolean statement logic.

27 $*$*$ (A6) ENCOUNTERED ERROR $*$*$

Agreement is link name of module which encountered an error.

28 NUMBER OF ADVERBS/VERB EXCESSIVE

Verb has more than 1000 clauses.

29 $*$*$ ILLEGAL ACTION (A6) $*$*$

Value is action stored in /ERRCOM/ block. (Legal actions are: "ABORT", "FLAG", and "PASS".)

30 SYNTAX ERROR - ILLEGAL ADVERB

Adverb not legal with input verb.

Figure 5. (Part 4 of 4)

4.4 BOOT Module

4.4.1 General Purpose. The BOOT Module is designed to create and update those portions of the data base which are essential to normal COP operation. As a result, the input to BOOT is on fixed formatted card images rather than a free form text English input. The portions of the data base which BOOT addresses are:

- The data organization index, which contains a functional description of the IDS data base structure, the information required to retrieve headers and the data editing directory
- The data entry point headers
- The dictionary
- The text English syntax directory
- The module link table

In general, each input card image to BOOT creates or updates a record within one of the above structures. Normally, BOOT will be run only when a QUICK data base is being created from scratch.

4.4.2 Input. The input to BOOT consists of an introductory command verb followed by a series of sets of card images. Each set is begun by a command adverb and terminated by a card containing END in card columns 1-3. The last set is followed by a second END card. The order of the sets is important if the user is building a data base from scratch because the creation of some records is dependent upon the previous creation of others. The order to be followed is that in which the sets will be presented.

4.4.2.1 Verb Command. The introductory verb for BOOT is INITIALIZE. This command must be defined starting with column 1 on the first card image that the COP reads in.

4.4.2.2 Introductory Adverbs. As cited, each set is introduced by a command adverb and ended with an END card. The command adverbs must appear on separate card images with the adverb starting in column 1. The command adverbs and the general description of the sets they introduce are shown in table 12. Discussions of each set follow.

4.4.2.3 NEWINDEX Adverb. This command adverb has no following cards. It must appear only in the case of a construction of the data base from scratch. In this case it must appear first. It causes the creation of the data organization index header and a utility table header used internally by COP.

Table 12. BOOT Command Adverbs

NEWINDEX	- Creates new index header
RECORDTYP	- Adds new record type records
HEADER	- Adds data entry point headers
DICTIONARY	- Makes entries and updates to the dictionary
MODULE	- Makes changes to the module link table
SYNTAX	- Makes entries and changes to the text English syntax directory
INDEX	- Makes entries and changes to the data organizational index

4.4.2.4 RECORDTYPE Adverb. A complete and total definition of each record type contained within the integrated data base is presented in Program Maintenance Manual, Volume I. This command adverb introduces the list of those record types to be added to the data organizational index. Each card image following the adverb contains the name and number of the record type. For each card image, the record type name is placed in columns 1 to 8 (left justified) and the record type number in columns 9 to 16 (free field). The set is ended normally with the insert of the END card.

4.4.2.5 HEADER Adverb. In the IDS sense, a header is the highest level of entry into a series of master records and their associated chains. In the QUICK data base, headers define entries to data elements through the use of attribute CLASS. This set of commands, then, specifies header name (or class entries) and, further, links them to record type as outlined above.

Each card in the set defines the record type name (columns 1 to 8, left justified), the attribute CLASS name (columns 9 to 16), the attribute SIDE name (columns 17 to 24). SIDE may be left blank. Also, if the header record type is TGTHD (target header), the card image may contain an entry for ICLASS (columns 25 to 32) which is used as an internal sequential counter. This set is ended by an END card.

4.4.2.6 DICTIONARY Adverb. All of the words that have meaning to the COP are contained within the dictionary (see appendixes A, B, and C) and are added or altered with this command adverb. Note then when the dictionary is altered, the corresponding INDATR record must be input to the BOOT run to cause a new INDATR IDS record (which contains common block address and attribute number) to be generated. The card image format is shown in figure 6. The word types defined with the dictionary are:

- OPERAT - text English mathematical, relational, boolean or syntax operator
- VERB - text English verb
- ADVERB - text English adverb
- SPCIAL - text English special word
- ATTRIB - attribute
- NULL - text English null

If the word is an attribute, the various type identifications are:

- NMALPH - alphabetic attribute
- IDALPH - alphabetic attribute used as an IDS identifier
- NMNUMR - numeric attribute
- IDNUMR - numeric attribute used as an IDS identifier.

Columns	Meaning
1-6	=INDATR
9-24	Attribute Name (left justified)
25-32	Attribute type mnemonic (left justified)
33-40	Default value (alphabetic values should be left justified
41-48	Lower limit (free field)
49-56	Upper limit (free field)

d. ALPHVL -- This card image creates the list of legal values for attributes whose type mnemonic is some combination of 'LIST' values. Each card image is:

Columns	Meaning
1-6	=ALPHVL
9-24	Attribute name (left justified)
25-32	'LIST' values (left justified)

e. LINKER -- This card image is used to inform the data organization as to the make up of each record type. The card image is:

Columns	Meaning
1-6	=LINKER
9-16	Record type name (left justified)
17-32	Attribute name (left justified)
33-40	Control identifier mnemonic (left justified) =CNTROL, attribute used as a key for internal structure definition =NORM, attribute not a control one

4.4.2.11 Final BOOT Card. After the last input set to BOOT has been terminated, the user must include another END card. Normal text English commands may follow.

4.4.3 Output. The BOOT module produces an annotated list of its input on print report code 13. Each card image as input is displayed as input. (Columns 73-80 are also printed so as to provide any sequencing information the user desires to place there.) Command adverbs are given a special flag. In addition, each card image is preceded by a +, @ , or $ symbol. Respectively, these symbols imply a record type was added, altered or an error occurred. A sample of portion of the BOOT module output appears in figure 7.

```
①***** CENTRAL OPERATIONS PROCESSOR *****

②INITIALIZE

③***** COLD BOOT PROCESS *****

④***** SYNTAX
⑤@ PRMADV  OMITTING        BOOL    ELEMENT
⑥$ SYNCLZ  DESIGN          ONPRINTS

④***** DICTIONARY
⑦+ SIZE            ATTRIB  338     0       NMNUMB

④***** END
```

HEADING	MEANING
①	COP input print header
②	Command to run BOOT .
③	BOOT Input Print Header
④	BOOT Input set introductory card
⑤	This card caused BOOT to alter a record
⑥	This card was rejected by BOOT
⑦	This card caused BOOT to add a record

Figure 7. BOOT Module Output Print

SECTION 5. DATA MODULE

5.1 General Purpose

The DATA module is designed to allow the user to create those portions of the data base not created by other modules, add information to existing record types which were only partially initialized by other modules, make corrections and updates to existing record types and delete record types which are no longer desired. The DATA module has three verbs: CREATE, CHANGE, and DELETE. The general philosophy of the DATA module is to allow the user to make commands solely in terms of attributes, relieving him of the necessity of knowing which record type or types are affected. Also, the execution of the DATA module is permissible at any stage of processing within the QUICK system.

5.2 Input

5.2.1 The CREATE Verb. The CREATE verb defines new data record types. The DATA module builds record types to be created filling in any unsupplied attribute values either from QUICKs directory defaults or from a user specified source. In general, the use of the CREATE verb will cause one or more sets of record types to be created. DATA also will check all input values against QUICKs directory for validity. Finally, the DATA module checks to make sure that no duplicate records are created such as two separate target records having the same DESIG.

If the user desires to create targets via DATA he does not necessarily have to specify values for attributes TYPE, DESIG, TASK, or any other attribute value normally obtained if the JLM has successfully built an Assignment table (see Users Manual Volume II, section 2). These values will be supplied from the Assignment table where not user specified. (Note that under certain options JLM can run successfully without an Assignment table. In these cases the user must supply all data and is responsible for the uniqueness of the value for the DESIG attribute.)

The CREATE verb has five legal adverbs:

- o SETTING - defines values to be stored
- o SAME - alternate method of storing default attribute values
- o USING - specifies the use of QUICKs directory
- o SUPRESSING - specifies that QUICKs directory is not to be used
- o ONPRINTS - optional request for printing data of interest to a maintenance programmer

A sample CREATE verb input could be:

```
CREATE SETTING CLASS=BOMBER SIDE=BLUE
(IREG, LAT, LONG) = (1,650000N,1500000W) & (2,630000N,0510000W)
(3,451015N,0900005W) ONPRINTS USING DIRECTORY
```

This command creates three records for the given CLASS and SIDE where each record has the input values for IREG, LAT and LONG. Note that TYPE, TASK, and CNTRYLOC are not given; they are assumed to be obtained from a created Assignment table.

Comments on the major use of adverbs follows.

5.2.1.1 The SETTING Adverb. This is the chief clause for the CREATE verb. It is that part of the command structure whereby the user informs the COP what values are to be stored and under what record type. Any attribute may be set through this adverb. If either LAT or LONG are to be set, their values must be input in DDMMSSH/DDMMSSH format, respectively. One setting clause will create one record type except if the clause contains one or more extended equals phrases (i.e., IREG=4 & 5 & 6) in which case a record type set will be created for every combination of the values in such phrases. Note that a create verb may contain any number of SETTING clauses.

5.2.1.2 The SAME Adverb. This clause allows the user to specify a logical record which is used to define the default values for attributes which the user has not input. Its purpose is to reduce the inputting of redundant information. For instance, say the user desires to create two weapon systems, one for a B52 and another for a B52A and both systems have identical attribute values except that the attribute RANGE differs between the two. Under this scenario, the user would have to input two complete sets of identical attribute values if not for the SAME clause. Assume the B52 record has been created and all of its attributes stored. A command, then, to create the B52A record could be:

```
CREATE SETTING TYPE=B52A RANGE=9000
       SAME AS TYPE B52
```

Now the B52A record type is created identically to the B52 but with a different value for RANGE.

It is appropriate to show examples of the LIKE and OF operation which are helpful in creating data. Consider the command:

```
CREATE SETTING TYPE=B52A RANGE LIKE TYPE F111
       SAME AS TYPE B52
```

Type B52A will have all of the same attribute values as the B52 except that the RANGE for the B52A will equal the range of the F111.

An example using the OF operator with the CREATE command would be:

CREATE SETTING CEP=10*CEP OF TYPE B58 . . .

The CEP to be created equals 10 times the CEP value stored for TYPE B58.

The general form for the SAME adverb is:

SAME AS Identifier attribute Identifying value

The combination of attribute and value is used to retrieve a record of the type defined in the SETTING clause where the identifier attribute has the identifying value. The attribute defaults are then retrieved. The attribute must be an attribute identified in the dictionary as being a unique identifying attribute (e.g., CLASS, TYPE, etc.). If this clause is omitted, the default values used for attributes not provided values in the SETTING clause will be those defaults found in QUICKs directory.

5.2.1.3 The SUPPRESSING and USING Adverbs. These adverbs are used to specify whether or not the DATA module is to use QUICKs directory to edit input values. If not specified, USING is assumed and all values are edit checked.

5.2.2 The CHANGE Verb. The CHANGE verb is used when the user wishes to alter existing record types. Information following the verb must be sufficient to inform the COP of what record type(s) is to be affected and what attribute values are to be changed. A WHERE clause is used for specification of the record types to be changed and a SETTING clause gives the altered attribute values. In addition to the standard optional ONPRINTS adverb, these are the only two adverbs recognized by the CHANGE verb.

Consider the command:

CHANGE WHERE CLASS=MISSIL AND SIDE=BLUE
AND CATCODE=2763&2764&2765 SETTING ISITE=1&2&3

Note that in this command both the WHERE and SETTING clause contains an extended equals phrase. In this case, DATA will assume that there is a one-to-one correspondence between the records which satisfy each element of the extension in the WHERE clause and the change specified in the same position in the SETTING clause. In general, every record type combination which satisfies the WHERE clause will be changed to match the values given in the setting clause. In the example, the MISSIL whose CATCODE equals 2764 would have its ISITE set to 2.

It should also be noted that it is a simple matter for DATA to handle a CHANGE verb whose WHERE clause contains only an equals clause (extended or not) and the attribute is DESIG.

A helpful hint in the use of CHANGE verbs is the knowledge that the COP processes all verbs sequentially. For a whole series of CHANGE verbs could be used where more than one of the verbs alters the same record type but setting different attribute values.

5.2.3 The DELETE Verb. The DELETE verb is used when the user wishes to eliminate a particular record of a certain type. The user should note that deletion of records from an IDS data base can have far reaching consequences. The DELETE verb has two legal adverbs: ONPRINTS and WHERE. In general, after determining the set of record types to be retrieved according to the WHERE clause, DATA retrieves these record types and deletes the type that is most subordinate in the data organization.

For example, the DELETE verb sample in the sample below will cause deletion of all refuel points in region 1.

```
DELETE WHERE CLASS=REFUEL&IREG=1
```

5.3 Output

5.3.1 Standard Output. The DATA module has standard output listed under the heading of "DATA MODULE QUALITY CONTROL", which is written on FILE CODE 13.

This standard report lists each data transaction sequentially. It also lists the type of transaction (CREATE, CHANGE, DELETE), the number of records affected by the transaction, the side affected (BLUE or RED) and the individual record types affected.

Zeros appearing beneath the column headings # RECS AFFECTED or REC TYPES AFFECTED indicates that the transaction was not completed and the user should investigate both the data base and the correctness of his data transaction.

5.3.2 Optional Output.

5.3.2.1 The CREATE Verb. The optional output of the CREATE verb consists of, for each supplied setting clause, a display of the retrieval scheme plus a message each time a record is created specifying its type. A sample of CREATE Verb output appears in figure 8.

5.3.2.2 The CHANGE Verb. The optional output of the CHANGE verb consists of a display of the retrieval scheme used plus a message each time a record is changed specifying its type. A sample appears in figure 9.

5.3.2.3 The DELETE Verb. The optional output of the DELETE Verb consists of a display of the retrieval scheme used plus a message each time a record is deleted specifying its type. A sample appears in figure 10.

5.3.2.4 Retrieval Scheme. The concept of data retrieval schemes (figures 8, 9, and 10) is identical for all verbs for this and other modules. Therefore, only one explanation is required and is as follows.

A COP retrieval scheme is a series of numbers which may be viewed as instructions. Each instruction consists of an introductory word which contains an identifying number, followed by one to three words which make up the remainder of the instruction. There are four instruction types: Get Header, Chain Next, Chain Master, and Return.

1 SETTING CLAUSE WRONG FOR CREATE

An error has occurred in the setting clause, check inputs.

2 (F15.4) VIOLATES EDIT RANGE FOR (A12)

Edit error for floating point attribute. Value and attribute name are displayed.

3 (I10) VIOLATES EDIT RANGE FOR (A12)

Edit error for integer attribute: value and attribute name are displayed.

4 (A6) IS NOT IN EDIT LIST FOR (A12)

Edit error for alphabetic attribute: value and attribute name are displayed.

5 DATA//ERROR IN SAME AS CLAUSE

Data could not find record identified in SAME clause.

6 ERROR IN TARGET INPUT

Error detected by Assignment table. Not sufficient amount of data supplied or the wrong data supplied.

7 LINKUP FAILURE

User supplied insufficient data to build a viable retrieval scheme.

8 DUPLICATE STORE ATTEMPTED (2A7,2I10)

Fields are record type name, DESIG, IREG and ICOMPL.
Message occurs when store attempt results in a D01 error--this error is not fatal.

9 DATA//HEADER NOT FOUND FOR (A6)

Value is illegal value entered for CLASS.

10 SAVLAT OR SAVLON ARRAY IS OVERFLOWING

More than 200 LATS or LONGS were input in one SETTING clause. Edit checking will be valid only for the first 199 LAT/LONGS.

Figure 11. CREATE Verb Error Messages

1	ERROR IN CHANGE VERB - MISSING CLAUSE CHANGE verb requires both WHERE and SETTING clauses.
2	SETTING CLAUSE WRONG Error in SETTING clause, check inputs.
3	WHERE CLAUSE WRONG Error in WHERE clause, check inputs
4	NO RECORD TYPES DETERMINED IN CHANGE User has not specified sufficient data to build a retrieval scheme.
5	LINKUP FAILURE Same as above.
6	(I10) VIOLATES EDIT RANGE FOR (A12) The first value is input as change for attribute shown as second value. Value violates directory limits. Input value is honored, however.
7	(F15.4) VIOLATES EDIT RANGE FOR (A12) Same as above for a floating point attribute.
8	(A6) IS NOT IN EDIT LIST FOR (A12) The first value is not in the directory's list of valid values for the attribute shown as the second value.
9	ERROR IN WHERE CLAUSE QUEUE Attribute in where clause queue also appears outside of the queue in the same clause.

Figure 12. CHANGE Verb Error Messages

SECTION 7. REPORT MODULE

7.1 General Purpose

The purpose of the REPORT module is to give the user/analyst the capability to produce ad hoc print reports. REPORT is capable of addressing any set of attributes in the QUICK integrated data base and displaying them, plus any arithmetic calculations upon them, in virtually any format the user desires. REPORT is called by three verbs: DESIGN, PRINT, and ALTER.

The DESIGN verb is used to specify the format and the sort to be used to display the data, the calculated variables to be included and the subset of the data base to be used. Further the user may give the display a name so that it is retained in the data base.

The PRINT verb produces the display the user has designed. In addition, the user may specify a different data base subset with the PRINT verb.

The ALTER verb is used to make changes to an existing saved 'display' or to construct a new 'display' from a previous one.

7.2 Input

7.2.1 The DESIGN Verb. The DESIGN verb may have the following adverbs: DISPLAY, SETTING, SORT, WHERE, DEFINE, FORMAT, ONPRINTS. The general command form is:

```
DESIGN [DISPLAY display-name [ ( OLD ) ]]
                             ( NEW )

       [SETTING ( PAGELENGTH ) value ]
                ( LINELENGTH )
                ( REPORTCODE )

       [SORT ( variable ) ( ASCENDING ) or ( A ) ]
             ( attribute ) ( DESCENDING )   ( D )

       [WHERE where-clause]

       [DEFINE variable ( = ) value-expression]
                        ( EQUALS )

       [FORMAT format-clause ]

       [ONPRINTS]
```

Section 3 detailed all of these adverbs. However major points concerning these adverbs will be highlighted within subsections to follow. Also within the discussion to follow (and for sections on the ALTER and PRINT verbs) reference will be continually made to a sample input as given in figure 18.

7.2.1.1 The DISPLAY Adverb. The DISPLAY clause is used to identify the display for future use. If it is omitted, the module will assign a general name. The clause consists of a display name of up to 12 characters and the option of either of the keywords OLD or NEW. If OLD appears, the module will replace the old display (stored within data base) with the new one. If NEW, the module will assure that no duplication of names occur. The default is NEW.

In the example (figure 18) the display name is set to TEST and is replacing an old display. All adverbs, then, that follow are 'owned' by the display name TEST.

7.2.1.2 The SETTING Adverb. The SETTING adverb as used with the DESIGN verb is used to specify non-default values for length of print page, line length, and print file unit. The only three attributes allowed are:

- PAGELENGTH - The length (or number of lines) of the physical page for output. (default=55 lines)
- LINELENGTH - The length (or number of characters) of the physical line for output. (Default=120 characters)
- REPORTCODE - The print unit used for output (default=42, which results in report code 52 (decimal 42))

Referring to figure 18, defaults are used, except that only 15 lines are permitted per page.

7.2.1.3 The SORT Adverb. The SORT clause (see section 3) is used to specify the order in which records involved in the display are to be processed. The clause consists of one or more pairs of items. The first item is either an attribute found in the desired records or a variable calculated from the record. The second item of each pair specifies the order of the sort as either ASCENDING (or A) or DESCENDING (or D).

Figure 18 shows that the records to be printed will appear according to increasing range values.

7.2.1.4 The WHERE Adverb. This clause is used to subset the data base. The clause consists of a boolean statement whose elements are relational phrases which may refer to attributes or any normal defines (see below). It may not contain any attribute using the OF phrase.

The example informs REPORT that only those records on the BLUE side are of concern.

SECTION 9. EXTERNAL INTERFACE MODULE (EIM)

9.1 General Purpose

The purpose of the EIM is to create output tapes/files which are designed to be input to external processors. EIM has three command verbs: BUILD, PLOTDATA and PLOTIT.

The BUILD verb has the capability to create two standard tapes plus a generalized output capability similar to that of the REPORT module. The user specifies the output file to be developed by inserting the special word SIDAC, TABLE, or OTHER within the BUILD command. The SIDAC word causes the production of two JAD format data base assessment tapes. The TABLE word causes the production of a standard set of tables. The OTHER word provides the capability to produce generalized data files.

The PLOTDATA and PLOTIT verbs produce an output tape suitable for use on the CALCOMP plotter. The plotable information includes penetration and depenetration corridors, refuel points and recovery bases where PLOTDATA is concerned, and bomber and tanker sorties where PLOTIT is used.

9.2 Input

9.2.1 The BUILD Verb. The BUILD verb has six optional adverbs: UNIT, WHERE, SORT, DEFINE, FORMAT, and ONPRINTS. It also has one required adverb FILE. The general form is:

BUILD FILE { TABLE / SIDAC / OTHER }

. . . [ON UNIT numeric-value]

. . . [WHERE where-clause] . . . [SORT sort-clause]

. . . [DEFINE define-clause]

. . . [FORMAT format-clause]

. . . [ONPRINTS]

9.2.1.1 FILE TABLE. This clause may be accompanied by a UNIT, WHERE, and/or ONPRINTS clause. The standard output unit defaults to 35.

9.2.1.2 FILE SIDAC. This clause stands alone except for the optional ONPRINTS clause. BLUE targets are defined on unit 35 and RED targets on unit 36.

9.2.1.3 FILE OTHER. Generalized data files are produced through user inputs. A FORMAT clause must be used and UNIT, WHERE, SORT, DEFINE, ONPRINTS may be included.

9.2.1.4 The UNIT Adverb. The UNIT clause is used to specify an output unit other than 35.

9.2.1.5 The WHERE Adverb. For FILE TABLE or SIDAC instructions, this clause may only be used to specify a value for the SIDE attribute. If omitted the value of SIDE defaults to BLUE.

For FILE OTHER this clause is used to specify the subset of the data base to be accessed to build the output file. This clause is fully detailed in section 3 and examples of usage within the REPORT module, which is similar to the EIM, are presented in section 8. Normal DEFINE variables may be used. However, the OF phrase is not permitted.

9.2.1.6 SORT and DEFINE Adverbs. The use of these clauses are as given for the REPORT modules (section 8).

9.2.1.7 The FORMAT Adverb. The use of the FORMAT clause is different in the EIM from REPORT (section 8) in that headers, trailers and PAGNO must not be used. Furthermore the PAGE special word has a different meaning than explained elsewhere. When it encounters the word PAGE, the EIM writes an end-of-file mark on the output unit. (The user should declare the output tape as a multi-file.)

9.2.2 The PLOTDATA Verb. The PLOTDATA verb has one adverb SETTING. This adverb is followed by a series of equal relational phrases. Each phrase sets the values for a particular portion of the plot information. The general form of the PLOTDATA verb is as follows:

PLOTDATA SETTING

... MAP { EQUAL / = } { PIC1 / PIC2 / PIC3 / PIC4 / PIC5 }

... SIDE { EQUAL / = } { side / BOTH } [AND side]

... SCALE { EQUAL / = } numeric-value

... (LAT, LONG) { EQUAL / = } (latitude, longitude)

Table 15. (Part 3 of 6)

WEAPON CHARACTERISTICS LIST

Column	Meaning
1-8	'F1WEAPON'
9	Side: 1 for Blue; 2 for Red
10-14	Line count
15-17	Blank
18-19	Warhead type (Line count * 10 plus 1 for ASM; 0 for all others)
20	0 = Bomb, 1 = ASM, 2 = DECOY
21-37	Blank
38-43	Warhead yield in kilotons
44-46	Fission to Fusion percentage (FFRAC*100)
47-90	Blank

Table 15. (Part 4 of 6)

MISSILE BASE LIST

Column	Meaning
1-8	'F1MIBASE'
9	Side: 1 for Blue; 2 for Red
10-14	Line count
15	Blank
16-20	Index Number (INDEXNO), numeric
21	Blank
22-28	Latitude (LAT) degrees, minutes, seconds; S if South, N if North
29-36	Longitude (LONG) degrees, minutes, seconds; E if East, W if West
37	Blank
38-41	Vulnerability Number (VULN1) alphanumeric
42-43	Type Count
44-45	Blank
46-47	'1/'; that is, beginning sortie number (always 1) followed by /
48-49	Number per site (NMPSIT), numeric
50	Blank
51	H if VN greater than or Equal to 20, S otherwise
52	Blank
53	1 if column 51 is H or if 51 is S and NOALER Equal NMPSIT Otherwise = 2
54-59	Blank
60-65	NAME, alphabetic
66-69	Blank
70-71	Country Location (CNTRYL), alphabetic
72-74	Blank
75-80	TYPE, alphabetic
81-84	Blank
85-90	BENO

1	ERROR IN FILE CLAUSE FILE clause contains error or is absent
2	ERROR IN WHERE CLAUSE Attribute is not SIDE (TABLE only)
3	ERROR IN SETTING CLAUSE Illegal attribute or value (PLOTDATA only)
4	POINT (I3) OFF MAP X = (F10.3) Y-10 = (F10.3) - point coordinates exceed plot limits (PLOTDATA and PLOTIT only)
5	ERROR IN FORMAT CLAUSE Probable cause is illegal special word (OTHER only)
6	ERROR IN DEFINE CLAUSE DEFINE clause contains an error (OTHER only)
7	DEFINES CANNOT BE RESOLVED No order can be found in which to properly execute define variable calculations (OTHER only)
8	ERROR IN SORT CLAUSE Sort parameters illegal, missing or in the wrong order (OTHER only)
9	LINKUP ERROR IN BLDOTH WHERE clause is not definitive enough to build a retrieval scheme. Put in more attributes to force scheme to desired record (OTHER only)

Figure 27. EIM Error Messages

Attribute Name	Description
ATTRSU	Penetration corridor attrition parameter after defense suppression
ATYPE*	Legal type assignment in Assignment Table
BENO	Bomber Encyclopedia Number
BPENFAC	Multiplier for bomber attrition parameters
CAPACITY	Vehicle capacity for a recovery base
CATCODE	Target category code
CATHI*	Upper limit of category for type assignment
CATLO*	Lower limit of category for type assignment
CCREL	Command and control region reliability
CEP	Circular Error Probable, delivery error applicable to bomber and missile delivered weapons (nautical miles) (for missile - CEP at maximum range)
CEPASM	CEP for an air-to-surface missile at maximum range
CEPMIN	CEP for missile at minimum range (RNGMIN)
CHAINNAM*	Chain name in data structure index
CLASS	Class name assigned to identify various subsets of the data base
CLAUSESW*	Switch that identified verbs that require clauses
CLAUSETY*	Value that identifies an adverb's clause type (Boolean, Sequential, Single or Null)
CLOSE	Determines magnitude of closing force in allocation process
CLOSER	Controls rate of increase in closing force
CMISS	Constant used in missile time-of-flight calculation
CNFLG*	Flag specifying country owner or location restriction in Assignment Table
CNTRYLOC	Country code for country where target is located
CNTRYOWN	Country code for country which owns the target
CORBOMB	Number of nautical miles prior to corridor entry
CORMSL	Fraction of missile flight completed at time zero
CORNUM	Corridor identification number
CORR2	SMAT array multiplier for final allocation evaluation
CORR	SMAT array multiplier
COUNTRY*	Legal country codes in Assignment Table

Attribute Name	Description
CPASMZRO	CEP for air-to-surface missile at zero range
DEFDIST1	Length of precorridor dogleg 1
DEFDIST2	Length of precorridor dogleg 2
DEFDIST3	Length of precorridor dogleg 3
DEFRAN	Typical range of interceptors at defense bases near a corridor (nautical miles)
DELTVAL	Maximum fractional difference in value allowed in allocators time-of-arrival cells
DESIG	Target designator code
DESIGA2*	Alpha portion of assignable DESIG in Assignment Table
DGZLAT	Complex lead or single point target latitude
DGZLONG	Complex lead or single point target longitude
DISPNAM1*	REPORT display name (first half)
DISPNAM2*	REPORT display name (second half)
DISTANCE	General distance attribute
DISTDF	Distance from target to end of depenetration corridor
DOGLEG	Corridor dogleg identification number
DSIDE	Defending side
ELEMNTTY*	For adverbs with element phrases -- type of element (e.g., operator, special words)
ELEMNTVL*	For ELEMNTTY of operator or special word, the exact operator or special word allowed
ERRCLOS	Controls termination of allocation processing
EXNBOMB	Number of vehicle 'loads' of weapons to be added to each bomber group

Attribute Name	Description
EXNMIRV	Same as EXNBOMB but for MIRVs
EXNMISS	Same as EXNBOMB but for single shot missiles
EXPASM	Fraction of bomber groups weapons which are ASMs
FACMIRV	Modifies SMAT array for MIRV systems
FFRAC	Fission fraction (fission yield/total yield)
FIXED	Switch to indicate if weapon assignment was user fixed
FIXOPT	Fixed assignment option
FLAG	Numeric code (1 through 99) used to impose allocation restrictions
FLMULT	Flag set in PREPALOC to indicate if a fixed weapon can be offset in ALOCOUT
FSALVO	Salvo number of weapon assignment
FSNSTVTY	Controls sensitivity of multiplier adjustment during later part of allocation
FULL1*	Flag set when KOUNT1 is at its maximum
FULL2*	Flag set when KOUNT2 if at its maximum
FULL3*	Flag set when KOUNT3 is at its maximum
FULL4*	Flag set when KOUNT4 is at its maximum
FULL5*	Flag set when KOUNT5 is at its maximum
FUNCTI	Operational application code for a weapon system (e.g., ICBM)
FVALT1 FVALT2 FVALT3 FVALT4 FVALT5	Fraction of target value remaining at T1, T2, T3, T4, and T5, respectively

Attribute Name	Description
FVULN1	Fraction of value of target in first hardness component
GBASE	Number of launch bases in weapon group
GFRASM	Fraction of weapons in group which are ASMs
GLAT	Latitude of group centroid
GLONG	Longitude of group centroid
GNVEH	Number of vehicles in group
GNWPNADJ	Number of weapons in group to be allocated (includes any overallocation)
GNWPNS	Number of weapons in group
GPKNAV	Single shot kill probability of weapon group against a naval target
GREFCODE	Group refuel code
GREFTIME	Group refuel time
GROUP	Group identification number
GRPFLG	Flag used to control weapon grouping
GSBL	Probability of group's survival before launch
GSBLREAL	Same as GSBL but not adjusted for overallocation

Attribute Name	Description
GSTART	Starting weapon index
GTYPE	Group weapon type
GTYPREFC	Reference code of group's weapon type record
GYIELD	Group yield (megatons)
HAZ	Lethal radius for air burst for first hardness component
HAZ2	Same as HAZ but for second hardness component
HGZ	Same as HAZ but for ground burst
HGZ2	Same as HZ2 but for ground burst
HIGHFAC	Upper estimate of terminal ballistic missile capability
HILOAT	The ratio of the low altitude attrition rate to the high altitude attrition rate
IALERT	Alert status; 1 = alert, 2 = nonalert
ICLASS	Class index assigned for game
ICOMPL	Complex index
IDHOB	Preferred height of burst indicator
IGIW	Indices of General Industrial Worth (dollars)
IMATCH	Controls method used to determine if a weapon allocation has met the required minimum destruction fraction specified by MINKILL
INDEXNO	Index of a target used during processing to identify the target
INITSTRK	Side initiating attack
IPENMO	Penetration mode; 1 = aircraft uses penetration corridor, 0 = penetration corridor not used
IRECMO	Recovery mode; 1 = aircraft recovery planned, 0 = aircraft recovery not planned
IREFUEL	Bomber refueling code
IREG	Index to identify geographic region
IREP	Reprogramming index (capability of missile squadron)
ISITE	Site number
IVERIFY	Controls operation of allocator after final allocation
KORR	Corridor used by assigned weapon
KORSTY	Parameter to adjust mode of corridor penetration

Attribute Name	Description
KOUNT1*	Highest numeric value for a unique DESIGA2 in region 1
KOUNT2*	Highest numeric value for a unique DESIGA2 in region 2
KOUNT3*	Highest numeric value for a unique DESIGA3 in region 3
KOUNT4*	Highest numeric value for a unique DESIGA2 in region 4
KOUNT5*	Highest numeric value for a unique DESIGA2 in region 5
LABEL#	Input; plot label selection switch
LAT	Latitude; Stored internally in degrees, input as degrees unless subject of CHANGE or CREATE SETTING clause, then DDMMSSH
LAW	Specifies form of damage law used for area targets
LBMBREF	Reference code of last bomber sortie record
LCHINT	Time between successive vehicle launches from the same base (missile or bomber) subject to the simultaneous launch condition
LINELENGTH#	Input; length of report print line
LMSLREF	Reference code of last missile sortie record
LONG	Longitude; Stored internally in degrees, input as degrees unless subject of CHANGE or CREATE SETTING clauses, then DDDMMSSH
LOWFAC	Lower estimate of terminal ballistic missile capability
MAJOR	Target major complex number
MAP#	Input; map type selection

Attribute Name	Description
MASDETNM*	Master or Detail Name in data structure index
MASDETNO*	Master or Detail Number in data structure index
MAXFRA	Maximum value of weapon resources to be used relative to target value
MAXKILL	Desired maximum damage expected for a target
MAXSAL	Maximum number of salvoes
MINCAP*	Lowest acceptable JAD capacity for type in Assignment Table
MINDAMAG	Minimum fraction of target value which must be destroyed by each weapon allocated
MINKILL	The required minimum damage established for a target
MINOR	Target minor compound number
MISDEF	Target ballistic missile defense
MYRECOV1	Depenetration corridor recovery base 1 (DESIG)
MYRECOV2	Depenetration corridor recovery base 2 (DESIG)
MYRECOV3	Depenetration corridor recovery base 3 (DESIG)
MYRECOV4	Depenetration corridor recovery base 4 (DESIG)
NADBLI	NLRTDB for initiative attack
NADBLR	NLRTDB for retaliatory attack
NALERT	Same as spelling NOALER
NALTDLY	Nonalert delay
NAME	Alphanumeric descriptor for any item in the data base
NAREADEC	Number of decoys per independent reentry vehicle for area BMD

Attribute Name	Description
NASMTYP	Number of distinct types of ASMs
NBASES	Total number of launch bases
NCCREG	Number of command and control regions
NCMS	Number of counter measures carried by vehicle
NCOMPLX	Number of target complexes
NDCNTYCD	Number of distinct country codes
NDECOYS	Number of decoys on a bomber or number of decoys per independent reentry vehicle for terminal BMD
NDEPCRD	Number of depenetration corridors
NFIXES	Number of fixed weapon assignments for group
NHRDCOMP	Number of hardness components
NLRTDB	Probability of destruction before launch (DBL) of nonalert vehicle
NLRTDL	Delay of nonalert vehicle before commencing launch (hours)
NMPSIT	Number of missiles per site
NOALER	Number of vehicles on alert status (also spelled NALERT)
NOINCO	Number of delivery vehicles on commission
NOPERSQ	Number of weapon vehicles per squadron
NPAYLOD	Number of payload tables
NPENCRD	Number of penetration corridors
NPNCRTY	Number of penetration corridor types (-1 implies mini-allocator is to be used)
NPRCRDEF	Number of precorridor defense doglegs

Attribute Name	Description
NPRSQ1	Number per squadron - scenario 1
NPRSQ2	Number per squadron - scenario 2
NPRSQ3	Number per squadron - scenario 3
NPRSQ4	Number per squadron - scenario 4
NRECOVB	Number of recovery bases
NREFUEL	Number of refuel points
NSAL1 NSAL2 NSAL3	Number of weapons in salvo for salvoed weapon type (numbers are packed nine per word)
NSFIX1 NSFIX2 NSFIX3	Number of fixed assignments in salvo (numbers are packed nine per word)
NTANKERB	Number of tanker bases
NTARGETS	Number of targets as seen by the allocator
NTIMCOMP	Number of time components
NTINT	Number of terminal BMD interceptors at target
NUMALOC	Number of assignments by allocator
NUMDBL	Number of aircraft destroyed before launch
NUMLOAD	Number of warheads of a type in payload table
NWEPGRP	Number of weapon groups
NWEPTYP	Number of weapon types selected by user
NWHDS	Number of warheads per independent reentry vehicle (missile)
OFFLAT	Offset distance from DGZLAT of weapon assignment
OFFLONG	Offset distance from DGZLONG of weapon assignment

Attribute Name	Description
ORLAT	Penetration corridor orientation point latitude
ORLONG	Penetration corridor orientation point longitude
PAGELENGTH#	Input; length of report print page (lines)
PAYALT	Bomber payload release altitude
PAYNAME	Payload table name (used in payload-weapon linking)
PAYTBLNM	Payload table name automatic linkage if WHD type used
PCTIW#	Scaling factor for IGIW calculations
PCTPOP#	Scaling factor for POP calculations
PDES	Probability that launch failure destroys missile
PDUD	Probability that warhead will fail to detonate
PEN	Penetration probability of assigned weapon

Attribute Name	Description
PENPROB	Penetration probability for group formed using module DATAMAKE
PEXBOMB	Fractional number of bomber weapons to be added by PLANSET
PEXMIRV	Same as PEXBOMB but for MIRVs
PEXMISS	Same as PEXBOMB but for single shot missiles
PFIW#	Scaling factor for IGIW calculations
PFPF	Probability of failure during powered flight
PFPOP	Scaling factor for POP calculations
PHRASETY*	Value that identifies an adverb's phrase type (relational, Equal or Like, Elemental)
PINC	Probability that a missile is in commission
PKNAV	Single shot kill probability of a weapon against a naval target (a value greater than zero restricts weapon use to naval targets)
PKTX	Probability of warhead kill by terminal BMD
PLABT	Probability of launch abort
PLOT#	Plot selection for PLOTDATA and PLOTIT
POP	Population value for U/I target (thousands)
PRABT	Probability of refuel abort
PRM	Controls value of quadratic premium
PROBHIGH	Probability that terminal defense is modified by HIGHFAC
PROBLOW	Probability that terminal defense is modified by LOWFAC
QUALITY	Controls extent to which STALL will attempt to refine allocation
RADIUS	Size descriptor for area targets (nautical miles)
RADPX	Probability of warhead kill by area BMD
RANGE	Vehicle range (nautical miles)
RANGEASM	Range of ASM
RANGEDEC	Range decrement for low altitude aircraft flight (high range/low range)
RANGEMOD#	Adjusted weapon group range
RANGEREF	Range of bomber with refueling
RATIOINT	Ratio of longest integration period used to theoretical

Attribute Name	Description
REGION*	Region assigned to country in Assignment table
REL	Reliability - probability that weapon system will arrive at target given successful launch
RELASM	ASM reliability
REPORTCODE*	Input; report code for REPORT module output
RETARGET#	Input parameter specifying missile may retarget
RINTPRD	Approximate ratio between rate of change of target weights between different integration periods
RNGMIN	Minimum range (nautical miles) for the missile type, used in computing flight times
RVAL	Relative value of weapon assignment
SALVO#	Input; salvo number for fixed assignment
SCALE#	Input; map scale
SCENARIO#	DBMOD input scenario selection
SCHANGE	Alphanumeric indicator for a changed sortie
SCUMSURV	Cumulative survival probability for sortie event
SDAMEXP	Damage expected as a result of sortie event
SDELAY	Delay time for sortie
SDELTIME	Time change during sortie event
SDEPEN	Depenetration corridor used by sortie
SETTLE	Control the number of passes at PROGRESS = .75
SEVCODE	Sortie event code. Identified type of event
SIDE	Item side name
SIMLUN	Maximum number of vehicle launches which can occur simultaneously from one base
SINDEXNO	Sortie launch base index number
SLAT	Sortie launch base latitude
SLOCATTR	Local attrition rate for sortie event
SLONG	Sortie launch base longitude
SLOW	Available low distance for bomber sortie
SLOW1	
SLOW2	
SLOW3	
SNSTVTY	Controls sensitivity of multiplier adjustment during early phases

Attribute Name	Description
SORTNO	Sortie identifier code number
SPDLO	Speed at low altitude (knots)
SPEED	Speed (knots)
SPEEDASM	Speed of ASM (knots)
SPLACE	Used with SEVCODE as modifier
SREFUEL	IREFUEL code for sortie
STALADJ	Determines extent to which STALL favors high unit profit versus efficiency in selecting weapons
STARFAC	Multiplier level of bomber defense for sortie
SVEHNUM	Sortie vehicle number
T1 T2 T3 T4 T5	Times of departure of first through the fifth value components of a target
TABCHAR*	Dictionary tab character
TARDEF#	Level of target defense
TARDEFHI	Level of local bomber defenses at high altitude
TARDEFLO	Level of local bomber defenses at low altitude
TARFAC	Multiplier level of terminal bomber defenses
TASK	SIOP table number
TGTMULT	Number of elements in a complex
TGTNUMB	Target index in target list as given to the allocator
TGTREFCD	Target IDS internal Reference Code (used in target list)

Attribute Name	/Description
TINTFAC	Multiplier level of terminal BMD
TOFMIN	Minimum flight time (minutes) for missile types used in computing flight time
TTOS	Total time on station (for a tanker) (hours)
TYPE	Alphanumeric designator (type name) to identify sets in the data base
VAL	Relative value of an item within its CLASS as established in the data base base by the user (also spelled VALUE)
VERBVAL*	Verb's identifying number
VONBASE	Number of vehicles per launch plus index of starting vehicle Packed word - Index number for starting point of non-alert vehicles and number of launch vehicles on base.
VOZ	Normalized target value; complexing key in INDEXER
VULN1	First hardness component of a target
VULN2	Second hardness component of a target
WACNO	Target World Area Code Number
WEPNAME	Subset of weapon type
WHOB	Preset HOB for weapon
WORDSTR1*	First half of word in dictionary
WORDSTR2*	Second half of word in dictionary
WORDTYPE*	Identifies dictionary word as to type (i.e., Attribute, Verb, etc.)
WORDVAL*	Dictionary word identifying number within type
YIELD	Yield (Megatons)

APPENDIX B

QUICK DATA BASE DIRECTORY

The QUICK data base directory consists of a list of all the attributes which can be used to describe the data items defined in the integrated data base. The information contained in the directory for each attribute includes:

a. The name of the attribute plus an indicator that defines the attribute as belonging to a logical collection. If the name appears alone it is a gaming attribute; if an asterisk (*) follows the name it is a non-gaming attribute; if a pound sign (#) follows it is text English input

b. The type of the attribute may be:

- Single - appears only within one record type. Input values are either LIST, INTGER, ALPHA, or FLOAT depending on the mode.
- Multiple - appears within more than one record type. Input values are either MLTLST, MLTINT, MLTALP, or MLTFLT depending on the mode.
- Control - same as multiple, also is used as a key for internal structure definition. Input values are either CNTLST, CNTINT, CNTALP, or CNTFLT depending on the mode.

c. The modes, or input/output conversions. These are standard FORTRAN formats plus a list which specifies a list of alphabetic entries.

d. The default value to be assigned the attribute when it is not defined for an item.

e. The attribute lower limit

f. The attribute upper limit

Attribute Name	Type	Mode	Default	Lower Limit	Upper Limit
ACLASS*	Single	List	Other	-	-
ACTIVE	Single	Integer	0	0	0
ADBLI	Single	Float	0	0	1
ADBLR	Single	Float	0	0	1
ADVERBVL*	Control	Integer	0	0	0
ALPLSTVL*	Single	Alpha	-	-	-
ALRTDB	Single	Float	0	0	1
ALRTDL	Single	Float	0	0	168
ALTDLY	Single	Float	0	0	99
ARRIVE	Single	Float	0	0	99
ASGHOB	Single	Integer	1	0	1
ASIDE	Single	Alpha	-	-	-
ASNTASK*	Single	Alpha	ZZ	-	-
ATDEFALT*	Single	Alpha	--	-	-
ATRNGHI*	Single	Alpha	999999	-	-
ATRNGLOW*	Single	Alpha	0	-	-
ATTINC	Single	Integer	0	0	9
ATTPOS	Single	Integer	0	0	9
ATTRBTYP*	Multiple	Integer	0	0	0
ATTRCD	Single	Float	0	0	1
ATTRCO	Single	Float	0	0	1
ATTRIBAD*	Multiple	Integer	0	0	0
ATTRIBN1*	Control	Alpha	-	-	-
ATTRIBN2*	Control	Alpha	-	-	-
ATTRIBNO*	Multiple	Integer	0	0	331
ATTRLE	Single	Float	0	0	1
ATTRPRE1	Single	Float	0	0	1
ATTRPRE2	Single	Float	0	0	1
ATTRPRE3	Single	Float	0	0	1
ATTRSU	Single	Float	0	0	1
ATYPE*	Single	Alpha	-	-	-
BENO	Single	Alpha	-	-	-
BPENFAC	Single	Float	1	0	0
CAPACITY	Single	Integer	0	0	9999

Attribute Name	Type	Mode	Default	Lower Limit	Upper Limit
GBASE	Single	Integer	0	0	0
GFRASM	Single	Float	0	0	1
GLAT	Single	Float	0	-90	90
GLONG	Single	Float	0	0	360
GNVEH	Single	Integer	1	1	999
GNWPNADJ	Single	Integer	1	1	999
GNWPNS	Single	Integer	1	1	999
GPKNAV	Single	Float	0	0	0
GREFCODE	Single	Integer	1	0	1
GREFTIME	Single	Float	1	0	1
GROUP	Control	Integer	0	0	999
GRPFLG	Single	Integer	0	0	999999
GSBL	Single	Float	0	0	0
GSBLREAL	Single	Float	0	0	0
GSTART	Single	Integer	0	0	0
GTYPE	Single	Alpha	--	--	--
GTYPREFC	Single	Integer	0	0	0
GYIELD	Single	Float	0	0	99
HAZ	Multiple	Float	0	0	0
HAZ2	Multiple	Float	0	0	0
HGZ	Multiple	Float	0	0	0
HGZ2	Multiple	Float	0	0	0

Attribute Name	Type	Mode	Default	Lower Limit	Upper Limit
HIGHFAC	Single	Float	0	0	0
HILOAT	Single	Float	.1	0	1
IALERT	Single	Integer	0	0	0
ICLASS	Single	Integer	1	1	15
ICOMPL	Control	Integer	0	1	9999
IDHOB	Multiple	Integer	0	0	0
IGIW	Single	Integer	0	0	99999
IMATCH	Single	Integer	0	0	0
INDEXNO	Multiple	Integer	1	1	99999
INITSTRK	Single	Integer	2	1	2
IPENMO	Single	Integer	1	0	1
IRECMO	Single	Integer	1	-1	1

Attribute Name	Type	Mode	Default	Lower Limit	Upper Limit
IREFUEL	Single	Integer	0	-5	20
IREG	Control	Integer	0	0	20
IREP	Single	Integer	0	0	5
ISITE	Single	Integer	0	-100	100
IVERIFY	Single	Integer	0	0	2
KORR	Single	Integer	0	1	999
KORSTY	Single	Integer	0	0	10
KOUNT1*	Single	Integer	0	0	999
KOUNT2*	Single	Integer	0	0	999
KOUNT3*	Single	Integer	0	0	999
KOUNT4*	Single	Integer	0	0	999
KOUNT5*	Single	Integer	0	0	999
LABEL#	Single	Alpha	--	--	--
LAT	Multiple	Float	0	-90	90
LAW	Single	List	POWER	--	--
LBMBREF	Single	Integer	0	0	0
LCHINT	Single	Float	0	0	999
LINELENGTH#	Single	Integer	120	0	0
LMSLREF	Single	Integer	0	0	0
LONG	Multiple	Float	0	0	360
LOWFAC	Single	Float	0	0	0
MAJOR	Single	Integer	0	1	999999
MAP#	Single	Alpha	--	--	--
MASDETNM*	Multiple	Alpha	--	--	--
MASDETNO*	Multiple	Integer	0	0	0
MAXFRA	Multiple	Float	10	0	10
MAXKILL	Multiple	Float	1	0	1
MAXSAL	Single	Integer	0	0	0
MINCAP*	Single	Integer	0	0	0
MINDAMAG	Single	Float	0	0	0
MINKILL	Multiple	Float	0	0	1
MINOR	Single	Integer	0	0	999999

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
271	CLOSE	261
272	CLOSER	262
273	CORR	263
274	CORR2	264
275	DELTVAL	265
276	ERRCLOS	266
277	FACMIRV	267
278	FIXOPT	268
279	FSNSTVTY	269
280	HIGHFAC	270
281	IMATCH	271
282	IVERIFY	272
283	LAW	273
284	LOWFAC	274
285	MINDAMAG	275
286	PKTX	276
287	PRM	277
288	PROBHIGH	278
289	PROBLOW	279
290	QUALITY	280
291	RADPX	281
292	RATIONT	282

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
293	RINTPRD	283
294	SETTLE	284
295	SNSTVTY	285
296	STALADJ	286
297	TARFAC	287
298	TINTFAC	288
299	FIXED	182
300	ASGHOB	208
301	FSALVO	185
302	WEPNAME	135
303	SLAT	290
304	SLONG	291
305	SREFUEL	292
306	SDELAY	293
307	SDEPEN	294
308	SVEHNUM	295
309	SLOW	296
310	SLOW1	297
311	SLOW2	298
312	SLOW3	299
313	SLOCATTR	300
314	SCUMSURV	301
315	SDELTIME	302

ATTRIBUTE NUMBER	ATTRIBUTE NAME	ATTRIBUTE COMMON BLOCK ADDRESS
316	SDAMEXP	303
317	SEVCODE	304
318	SPLACE	305
319	SCHANGE	306
320	LMSLREF	167
321	LBMBREF	168
322	SORTNO	289
323	SINDEXNO	206
326	CEPMIN	144
327	PENPROB	87
328	ATTPOS	44
329	ATTINC	46
331	CPASMZRO	132
333	GRPFLG	124

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
SELECT	17	Verb
ASTERISK	18	Verb
ALLOCATE	19	Verb
DGZSELECT	20	Verb
EVALUATE	21	Verb
FOOTPRNT	22	Verb
PLANOUT	23	Verb
PLOTIT	24	Verb
POSTALOC	25	Verb
DATAMAKE	26	Verb
MIRVDUMP	27	Verb
FRAMIS	99	Verb
ALPHAS	1	Adverb
RECALC	2	Adverb
ATTACKERS	3	Adverb
DEFENDERS	4	Adverb
DEFINE	5	Adverb
DISPLAY	6	Adverb
FIELDS	7	Adverb
FILE	8	Adverb
FIX	9	Adverb

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
FORMAT	10	Adverb
ONPRINTS	11	Adverb
OMITTING	12	Adverb
PLAYERS	13	Adverb
PRIORITY	14	Adverb
REPLACING	15	Adverb
SAME	16	Adverb
SETTING	17	Adverb
SORT	18	Adverb
SUPRESSING	19	Adverb
UNIT	20	Adverb
USING	21	Adverb
WHERE	22	Adverb
WITH	23	Adverb
VNOPTION	24	Adverb
KEEPING	25	Adverb
ORDER	26	Adverb
UICLASSES	27	Adverb
FLAGREST	28	Adverb
LOCREST	29	Adverb
MINRANGE	30	Adverb
MIRVREST	31	Adverb
PUNCH	32	Adverb

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
MODRANGE	33	Adverb
READMUL	34	Adverb
SMAT	35	Adverb
ABTAPE	36	Adverb
ACARD	37	Adverb
CCARD	38	Adverb
COUNTRIES	39	Adverb
FINDMIN	40	Adverb
EQUATE	41	Adverb
FUNCOM	42	Adverb
GAMETIME	43	Adverb
ICARD	44	Adverb
IF	45	Adverb
PLANE	46	Adverb
REEQUATE	47	Adverb
STRIKE	48	Adverb
TGTMOD	49	Adverb
WPNMOD	50	Adverb
MISTME	51	Adverb
MSLCOR	52	Adverb
FIXOFF	53	Adverb

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
A	1	Special Word
ASCENDING	1	Special Word
AFTER	2	Special Word
D	3	Special Word

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
DESCENDING	3	Special Word
HEADER	4	Special Word
IN	5	Special Word
LINE	6	Special Word
TABLE	7	Special Word
NEW	8	Special Word
OLD	9	Special Word
OTHER	10	Special Word
PAGE	11	Special Word
REMOVE	12	Special Word
REPLACE	13	Special Word
SIDAC	14	Special Word
SPACES	15	Special Word
X	15	Special Word
TRAILER	16	Special Word
ALL	17	Special Word
OWNED	18	Special Word
PAGENO	19	Special Word
ACOS	20	Special Word
ACOT	21	Special Word
ASIN	22	Special Word
ATAN	23	Special Word
C	25	Special Word
COS	26	Special Word
COT	27	Special Word
EXP	28	Special Word

DICTIONARY WORD	IDENTIFICATION NUMBER	GRAMMATICAL GROUP
G	29	Special Word
HHR	30	Special Word
KDAY	31	Special Word
KMON	32	Special Word
KYEAR	33	Special Word
N	34	Special Word
SIN	35	Special Word
UPDATE	36	Special Word
INCLUDE	37	Special Word
EXCLUDE	38	Special Word
TAN	39	Special Word

Table 18. Attributes Required for MISSIL, BOMBER, and TANKER Target Classes

ALL

ATTRIBUTE NAME	COMMENT
ADBLI	Set by user
ADBLR	Set by user
ALRTDB	Set by DBMOD
ALRTDL	Set by user
GROUP	Set by PLANSET
IREFUEL	Set by user
ISITE	Set by user
NADBLI	Set by user
NADBLR	Set by user
NLRTDB	Set by DBMOD
NLRTDL	Set by user
NOALER	Set by DBMOD
NOINCO	Set by DBMOD
NOPERSQ	Set by DBMOD
NPRSQ1	Set by user
NPRSQ2	Set by user
NPRSQ3	Set by user
NPRSQ4	Set by user

Table 18. Cont'd

ALL

ATTRIBUTE NAME	COMMENT
NUMDBL	Set by User
PKNAV	Set by User
VONBASE	Set by PLANSET
WEPNAME	Set by user

needed attributes as outlined in table 19 . This requires a CHANGE verb and examples of exact input are:

```
CHANGE WHERE CLASS=TNKWEP & SIDE=BLUE & TYPE='KC-135'
       SETTING TTOS=3
CHANGE WHERE CLASS=MSLWEP & SIDE=BLUE &
            TYPE='MM-IA' & 'MM-IB' & 'MM-III' & 'MM-II' & TITAN
       SETTING FUNCTI=ICBM ALTDLY=0 NALTDLY=0 NMPSIT=1 SPEED=12000
          (CEP,IREP,LCHINT,PDES,PFPF,PINC,PLABT,RANGE,REL,SIMLUN)=
          ( 1,1,1,.1,.08,.85, .1,4700,.70,5) &
          (.8,2,0,.1,.07,.90, .1,5500,.75,1) &
          (.6,4,1,.1,.05,.90,.08,6300,.79,5) &
          (.6,4,0,.1,.05,.90,.08,6300,.79,1) &
          ( 1,2,0,.2,.16,.80,.17,7200,.56,1)
CHANGE WHERE CLASS=MSLWEP & SIDE=BLUE &
            TYPE='POL-A2' & 'POL-A3' & POSEID
       SETTING ALTDLY=1 FUNCTI=SLBM NALTDLY=0 NMPSIT=16
               PDES=0 SIMLUN=1
               (CEP,IREP,LCHINT,PFPF,PINC,PLABT,RANGE,REL,SPEED) =
               ( 1,4,1,.10,.86,.12,1500,.6,6000) &
               ( 1,4,0,.08,.86,.12,2500,.7,8000) &
               (.7,5,0,.10,.83,.15,2500,.8,10000)
```

WARHEAD and PAYLOD Classes. The linkage of weapon systems to payloads and warheads is very precise and the order of creation should be as outlined. Warhead characteristics are defined according to the attributes shown in table 20. Class entries are defined as:

- o BOMB - gravity bombs
- o ASM - air-to-surface missile
- o RV - single shot reentry vehicle
- o MRV - multiple reentry vehicle
- o MIRV - multiple independent reentry vehicle
- o FACTOR - auxiliary loading factors

Command examples that create new warhead records are:

```
CREATE SETTING CLASS=BOMB (TYPE,YIELD,PDUD,FFRAC) = ('MK-5',1,,.02,.7) &
       ('MK-7',2,,.02,.7) AND ('MK-18',4,,.02,.7) SIDE=BLUE
       SETTING CLASS = ASM (TYPE,YIELD,PDUD,FFRAC,RANGEASM,RELASM,CEPASM,
       SPEEDASM) = (MNDDOG,1.5,.02,.7,200,,.9,1,,600,) SIDE=BLUE
        SETTING CLASS= RV (TYPE,YIELD,PDUD,FFRAC) = ('MK-5',1,,.02,.7) AND
       ('MK-18',4,,.02,.7) SIDE=BLUE
       SETTING CLASS=MRV(TYPE,YIELD,PDUD,FFRAC) = ('MK-12',2,,.02,.7)
       SIDE=BLUE
       SETTING CLASS=MIRV (TYPE,YIELD,PDUD,FFRAC,NWHDS) = ('MK-17',,1,02,,7,3)
        AND ('MK-20',,05,,02,,7,10) SIDE=BLUE
CREATE SETTING CLASS=FACTOR (TYPE,NCMS,PAYALT,NDECOYS,NAREADEC)=
       (FACT1,0,HIVAL,0,0)&(FACT2,1,HIVAL,0,0)&(FACT3,1,HIGH,0,0)&
       (FACT4,2,LOW,2,0)&(FACT5,1,HIVAL,2,0)&(FACT6,0,' ',2,2)
       SIDE=BLUE
```

```
CREATE SETTING CLASS=PENCOR SIDE=BLUE
     (CORNUM,ORLAT,ORLONG,KORSTY,HILOAT,DEFRAN,ATTRSU,ATTRCO) =
     (  1, 50,350,1, .2,250,.0003,  .001) &
     (  2, 50,350,1, .2,250,.0003,  .001) &
     (  3, 55,215,4, .2,250,.0001, .0003) &
     (  4, 60,230,4, .2,250,.0001, .0003) &
     (  5, 50,235,4, .2,250,.0001, .0003) &
     (  6, 40,243,4, .2,250,.0001, .0003) &
     (  7, 60,275,2, .2,250,0.   , .0006) &
```

For each corridor, doglegs start at the corridor origin and work outward. A line segment is identified through attribute DOGLEG. Each corridor will store line segments sorted on the attribute DOGLEG. Any number is permitted for DOGLEG entries. It is suggested that DOGLEG values be initially entered in multiples of 10 (10, 20, 30, etc.). This numbering will permit ease of redefinition if added line segments are defined after initial creation. A command that links DOGLEGs to corridors is:

```
CREATE SETTING CLASS=PENCOR SIDE=BLUE DOGLEG=10
       (CORNUM,LAT,LONG,ATTRLE)=
       ( 3, 620000N,1720000W,0.) & ( 4, 680000N,1610000W,0.) &
       ( 5, 400000N,1330000E,0.) & ( 5, 360000N,1240000E,0.) &
       ( 7, 730000N,0850000E,0.) & ( 8, 523000N,0110000E,.0005) &
       ( 9, 430000N,0180000E,0.) & (10, 430000N,0350000E,0.) &
       (11, 710000N,0300000E,0.) & (12, 600000N,0200000E,0.)
```

Depenetration corridor creation is similar as for penetration corridors except that links must be established between corridor exit and permissable recovery bases. Attributes MYRECOV1, MYRECOV2, MYRECOV3, and MYRECOV4 define the DESIG of the recovery bases where an aircraft may land if exiting a given corridor. No other action is required for linkage. Commands are:

```
CREATE SETTING CLASS=DEPCOR SIDE=BLUE
       (CORNUM,MRYECOV1,MYRECOV2,MYRECOV3,MYRECOV4) =
       (1,AB800,AB801,AB802,AB803) &
       (2,AB804,AB805,AB806,AB807) &
       (3,AB500,AB501,AB502,AB503) &
       (4,AB504,'0','0','0') &
       (5,AB505,AB506,'0','0')
```

```
CREATE SETTING CLASS=DEPCOR SIDE=BLUE
       (CORNUM,DOGLEG,LAT,LONG)=
       (1,1, 430000N,1380000E) & (1,2, 430000N,1380000E) &
       (2,1, 380000N,1330000E) & (2,2, 380000N,1330000E) &
       (3,1, 420000N,0290000E) & (3,2, 420000N,0290000E) &
       (4,1, 420000N,0400000E) & (4,2, 420000N,0400000E) &
       (5,1, 370000N,0550000E) & (5,2, 370000N,0550000E) &
       (6,1, 370000N,0720000E) & (6,2, 370000N,0720000E)
```

Refuel points are created through a simple definition of latitudes and longitudes.

Recovery Bases. Recovery bases may be identified by the user at any point after the targets which represent them are stored in the data base. The process is to create a record by specifying a value for CAPACITY. In the create statement the user must be aware of the necessity of adequately defining the type of the target along with its DESIG. This usually means that the type's name (TYPE) and location (CNTRYLOC) be supplied with the DESIG. An example is:

```
CREATE SETTING SIDE=RED CLASS=AIRFIELD TYPE=RECOV CAPACITY=40
       (DESIG,CNTRYLOC)=(AB516,IT) & (AB513,EG) & (AB514,EG) &
       (AB515,EG) & (AB512,PO) & (AB511,CZ) & (AB509,YG) &
       (AB510,YG) & (AB505,BG) & (AB506,BG) & (AB507,BG) &
       (AB508,BG) & (AB155,UR) & (AB149,UR) & (AB150,UR) &
       (AB151,UR) & (AB152,UR) & (AB146,CU) & (AB147,CU) &
       (AB148,CU) & (AB153,MX) & (AB154,MX) & (AB145,MX)
```

APPENDIX F

EXECUTION OF THE QUICK SYSTEM

This appendix presents the detailed characteristics of user procedures for executing the QUICK system, or more appropriately, the execution of the QUICK executive software, the COP. The capability enables the user to submit batch-mode jobs or to execute remote terminal jobs using the CARDIN system.

BATCH and CARDIN JCL

Both methods of execution require the same basic set of JCL cards. The required object decks are accessed by use of SELECT cards. Three basic modules (EXEC, QDATA, and INPUT) are always required. In addition, the utility library (file UL) and the I-D-S data base (file QD) must be designated for all QUICK executions.

In addition to the basic QUICK JCL, a $ SELECT card must be included for each module that is to be executed. These cards will be followed by a QUICK data file (I*) and other files and tapes needed for the execution. Table 22 shows the files required for each module. A typical set of QUICK JCL appears in figure 28.

A store or restore prior to or after the basic QUICK model execution can be accomplished by adding JCL to the runstream. This is accomplished by inserting a $ SELECT 634IDP00/PERFORM/(RESTORE or SAVE) followed by a $ QD file card, which identifies the I-D-S data base, at the desired location.

```
$    IDENT     1820510/10/6350,QUICK RUN JCL,C314
$    PARAM     15300
$    SELECT    634IDPOO/PERFORM/RESTORE
$    PRMFL     QD,R/W,R,634IDPOO/QUICKII/DATA
$    SELECT    634IDPOO/QUIK/COP/CANOF/EXEC
$    SELECT    634IDPOO/QUIK/COP/CANOF/DATA
$    SELECT    634IDPOO/QUIK/COP/CANOF/INPUT
$    SELECT    634IDPOO/QUIK/COP/CANOF/ALOC
$    SELECT    634IDPOO/QUIK/COP/CANOF/ALOCOUT
$    OPTION    FORTRAN
$    EXECUTE   DUMP,DEBUG,NREST,JREST
$    FILE      H*,HIS,75R
$    FFILE     P*,LGU/(06,42,43,11,12,13)
$    PRMFL     UL,R,R,634IDPOO/QUIK/LIBRARY/UTIL
$    PRMFL     PL,R,S,LIBRARY/PLOTLIB
$    PRMFL     QD,R/W,R,634IDPOO/QUIKII/DATA
$    DATA      I*
$    SELECT    634IDPOO/DATA/ALOC
$    SELECT    634IDPOO/DATA/ALOCOUT
$    TAPE9     15,T15D,,15301,,RECALC-TAPE
$    LIMITS    30,78K,-48K,30K
$    FILE      25,X25S,150R
$    FILE      21,X21S,50L
```

Figure 28. Quick Execution JCL (Part 1 of 2)

```
$       FFILE       21,NBUFFS/2,NOSLEW
$       FILE        22,X22S,50L
$       FILE        22NBUFFS/2,NOSLEW
$       FILE        23,X23S,50L
$       FFILE       23,NBUFFS/2,NOSLEW
$       FILE        24,X24S,50L
$       FFILE       24,NBUFFS/2,NOSLEW
$       PARAM       15302
$       SELECT      634IDPOO/PERFORM/SAVE
$       PRMFL       QD,R/W,R,634IDPOO/QUICKII/DATA
$       ENDJOB
```

Figure 28. (Part 2 of 2)

Remote Terminal Generation of JCL

A time-sharing capability exists which permits the user to generate the JCL needed to execute and/or compile QUICK modules. This is accomplished by an interactive question and answer activity at the terminal. JCL produced by this method may be executed directly using the CARDIN system or may be output as cards using the BPUNCH instruction for later submission as a BATCH job. Procedures to utilize this JCL generation function are outlined below. In the following narrative, commands to be entered by the user will be flagged by an arrowhead (→).

Step 1. Log On and Initiation. The first step is to log onto the HIS system and to specify what operating system is to be used and then what modules are to be executed. Steps involved in this example utilize a catalog file called 634IDPOO, RUNCOP which contains the procedures to execute COP. These steps are:

→ $*$LOG24,TSS

TERMINAL CK

USERID$PASSWORD

→ (enter userid and password)

IDENT?

→ (enter ident card information)

0 BLOCKS FILE SPACE AVAILABLE

CLASSIFICATION OF YOUR OUTPUT

→ (enter classification code, i.e., UZZ)

CLASSIFICATION OF FILES YOU WILL CREATE

→ (enter classification code, i.e., UZZ)

SYSTEM ?

→ RUNY 634IDPOO/RUNCOP

The user now is interacting with a YFORT TSS subsystem program. The YFORT program will signal its desire for a response by outputting an equal sign (=) to which the user responds accordingly.

Table 22. JCL File Utilization

Unit File Code	Comment
QD	Must be included; contains the COP I-D-S data base file
2	TGTFIL used by IIM
08	BASFIL used by IIM
15	Weapon data file for ALOC
17	Output file from MIRVDUMP
18	BTB Tape, JAD output (ASTERISK option)
19	TARFIL used by IIM, PLANSET
20	JAD input unit. Used by JLM.
21 22 23 24	Internal sort files. Used by: JLM, REPORT, EIM, EVALUATE, PLANSET
25	Random storage file used in concert with sort: JLM, REPORT, EIM, INDEXER
30	Directory file used by IIM
31	Spill tape used by IIM
32	Tape with I-D-S data base for SAVE or RESTORE (SRM)
33	Tape with I-D-S data base for SAVE or RESTORE (SRM)
35 36	Output tape files, used by EIM

After the system outputs an equal sign, the user has multiple responses in some cases. In order to assist the user, a response of HELP may be entered and the system will respond with all the possible replies. Permissible use of the HELP command is outlined below.

A new user will be presented with a series of questions and answers to initialize the required data files.

The first question the old user is asked is whether or not output is to be displayed at a terminal. The following message is displayed:

DO YOU WANT THE OUTPUT DIRECT TO JOUT? (YES OR NO)

A YES response will permit the use of SYSTEM JOUT. Otherwise, all output will be directed to the printer.

The user is then asked

DO YOU WANT TO USE AN URGENCY? (YES OR NO)

A YES response will add an urgency of (47) to the first line control card.

The user will then be given the opportunity to change the job name on the $IDENT card. The display requesting this information will be

ENTER JOB NAME OR NULL RESPONSE (8 CHARACTERS MAXIMUM)

Step 2. Mode Selection. Three basic modes of operation exist. These are execute COP (RUN), initialize the I-D-S data base (INITIAL), and compile QUICK source programs (COMPILE).

Any combination of modes is legal providing COMPILE precedes RUN or INITIAL and INITIAL precedes RUN. The response to statement

ENTER MODE (RUN, INITIAL, OR COMPILE).

Should be the first desired mode occurring in the sequence just discussed. For example, if a COMPILE and RUN is desired, the response should be

COMPILE

The reader should proceed directly to the step which discusses the initial input mode when following the step by step procedure.

Step 3. COMPILE. This mode allows the user to compile new or changed source code and to create object decks which can be selected to execute QUICK. The question

DO YOU WANT DUAL OR TEST CATALOG

allows the user to compile programs in the test catalog using a test version of common block C-30.

The next program response is

```
WHICH MODULES?
```

The names of the modules should be separated by commas with no imbedded blanks. HELP will produce a list of the legal modules and a repeat of the question.

A possible response would be:

```
→ DATA,JLM
```

Some of the modules have been subdivided into smaller sections for compilation purposes. If one of these modules (for example PLANOUT) is selected, the terminal response will be

```
PLANOUT  HAS  THE FOLLOWING  NEWCANOFS
PLANO    PLNT      INTR      TANK
WHICH NEWCANOFS DO YOU WANT TO COMPILE?
```

The desired response should be entered as follows

```
→ PLNT,TANK
```

If an unknown module is requested (either a new module or a misspelled module), the following will be displayed

```
PLAYOUT IS NOT A STORED MODULE

DO YOU WANT TO COMPILE THIS MODULE (YES OR NO)
```

A YES response will result in the generation of JCL to compile the module. Only four unknown modules can be added in a single run. In all cases, the user will be given an opportunity to add additional or corrected modules.

The final display in the compile mode is

```
DO YOU WANT TO RUN OR INITIALIZE AND RUN?

ENTER RUN, INITIAL OR NULL RESPONSE
```

If a reply of INITIAL is made, the reader should proceed to step 5. The RUN mode is discussed in step 6.

Step 4. Data Base Selection. Prior to completing the INITIAL and/or RUN steps, the desired I-D-S data base must be selected. If only one data base file exists on the users data file, that file will be selected and no response from the user is accepted. If more than one data base is on the users file, they will be printed. The user then inputs the desired I-D-S data base file.

Step 5. INITIAL. A file which is to be used as an I-D-S file must be initialized for the characteristics of that file prior to being used. INITIAL generates the JCL to accomplish this initialization. For more detail see section entitled, "Preparing an I-D-S File for COP" which follows.

Step 6.

Run-Save or Restore - The user can elect to restore or save the current I-D-S data base prior to the run. This can be accomplished by an appropriate response to the question

DO YOU WANT AN INITIAL RESTORE OR SAVE

ENTER RESTORE, SAVE, OR NO

If a save or restore is desired, the user is asked to input the tape number.

Input of Modules - The user is then requested to input the required modules. The three basic QUICK modules and any modules requested by the compile mode should not be entered. The format for entry is

→ POSTALOC,FOOTPRNT,PLANOUT

HELP can be entered to get a listing of the modules. The input modules are checked for proper spelling. If the input module is not stored, the user will be asked to verify that the unknown module should be executed.

Input of Verbs. The COP executes a wide variety of modules and differing size data bases which implies varying limits as to computer resource usage in terms of computer time, core storage, and lines to be printed.

A series of commands are given to provide interactive capability whereby the limits may be altered. First, the system must know which modules (or verbs) to execute. Hence the command:

WHICH VERBS

The verbs are separated by commas with no imbedded blanks. If HELP is entered, the list of valid verbs is displayed and the question is asked again.

A possible response is

→ CREATE,CHANGE,PRINT

The program responds with a set of default limits such as

LIMITS ARE 30,49K,,20K

NEW LIMITS? (ENTER NO OR ENTER NEW LIMITS)

=

If the displayed limits are valid, a NO response is entered. Otherwise, new limits are entered. For example:

→ 20,49K,,20K

Text English Command Entry. User inputs are now given. The display command is:

INPUT DATA FILE (ENTER CATALOG/FILE STRING OR DONE)

=

Here the program is asking the user to name a data file in which the user has stored text English commands. The user will be able to enter any number of such files, in the order as input. The user will also be able to enter text English commands directly and interweave files with direct entry by following this procedure. If the user enters the catalog file string of a file, the program will return to the preceding query. If the user has entered all the files names desired, DONE or a blank is entered.

The next message displayed is:

ANY ADDITIONAL DATA? (YES,NO)

A NO response will terminate the data input process.

The program now requests the possibility of text English commands entered directly by displaying:

ENTER DATA CARDS (BLANK LINE TERMINATES)

=

All inputs are entered followed by a blank line.

To allow for the possibility of additional data, the program displays;

ANY ADDITIONAL DATA FILES (YES,NO)

If YES is entered, the program again requests input data files. A NO response terminates the data entry.

Special Files. Certain verbs may require special input or output files. For each verb previously entered that may require a special file the following type of message is displayed:

THE SELECT VERB MAY NEED JAD FORMAT INPUT FILE

ENTER THE FILE USING ONE OF THE FOLLOWING FORMATS

(A BLANK OR NULL RESPONSE IF THE FILE IS NOT NEEDED)

$:FILE:20,F205,10L

$:TAPE9:20,T20D,,99999,,INPUT-TAPE

$:PRMFL:20,R,S,CATALOG/FILENAME

The proper file card should be entered based on the details of this particular run.

Step 7. Post - Run SAVE. After the completion of the run, the data base can be saved by answering

→ YES to the question

DO YOU WANT TO SAVE? (YES OR NO)

Step 8. Executing the COP. The constructed job may now be submitted which must be done within the CARDIN Subsystem of HIS. Applicable lines from the following are displayed:

THE JOB HAS BEEN BUILT TO { COMPILE / COP / COMPILE AND RUN COP }

AND DIRECT THE OUTPUT TO JOUT

```
WITH URGENCY U47

TO EXECUTE ENTER-

RUN THEJOB

READY

*

→ RUN THEJOB

SNUMB # XXXXX

*
```

The job has now been entered, the user should note the SNUMB.

Preparing an I-D-S File for COP

The COP must run on an I-D-S file. When the user wishes to restore onto a previously unused file or build a QUICK data base from scratch, he must use a file specially prepared for I-D-S. The creation of such a file has two steps. First, the user must create the file using the File Management Subsystem (FMS). Besides the usual options employed to create a random file, the following additional options are used:

BASESIZE/N/ - N defines the maximum number of pages in the I-D-S data file

RNG/r1,r2/ - Defines the page range (for QUICK applications set, r1 = 1 and r2 = N)

LINESPERPAGE/m/ - m defines the number of lines on an I-D-S page (for QUICK applications set m = 21)

An example of an appropriate FCREATE directive would be:

```
FCREATE/IDS/MYFILE, BASESIZE/401/,RNG/1,401/,LINESPERPAGE/21/,

SIZE/102/,MODE/RAND/,FCLASS/UZZ/,ABORT/OFF
```

Following the creation of the I-D-S file, the user must initialize the file by utilizing an I-D-S utility called QUTI. This utility has a

single input card that specifies the page range to be initialized. An example of an appropriate QUTI activity would be:

```
$    PROGRAM QUTI

$    PRMFL  A1,R/W,R,634IDP00/MYFILE

IDS  INITIAL 1,401
```

If the user needs any further assistance, consult Honeywell Reference DC53A, Rev.0 #I-D-S/I USER'S GUIDE.

DISTRIBUTION

Addressee	Copies
CCTC Codes	
C124 (Reference and Record Set)	3
C124 (Stock)	6
C126	2
C313	1
C314	7
C630	1
DCA Code	
205	1
EXTERNAL	
Chief, Studies, Analysis and Gaming Agency, OJCS ATTN: SFD, Room 1D935, Pentagon, Washington, DC 20301	2
Chief of Naval Operations, ATTN: OP-654C, Room BE781 Pentagon, Washington, DC 20350	2
Commander-in-Chief, North American Air Defense Command ATTN: NPXYA, Ent Air Force Base, CO 80912	2
U.S. Air Force Weapons Laboratory (AFSC) ATTN: AFWL/SUL (Technical Library), Kirtland Air Force Base, NM 87117	1
Director, Strategic Target Planning, ATTN: (JPS), Offutt Air Force Base, NE 68113	2
Defense Technical Information Center, Cameron Station, Alexandria, VA 22314	12
	42

AD-A086 160 COMPUTER SCIENCES CORP FALLS CHURCH VA F/G 9/2
THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM. (QUICK). USE--ETC(U)
APR 80 D J SANDERS, P F MAYKRANTZ, J M HERRIN DCA100-78-C-0042
UNCLASSIFIED CCTC-CSM-UM-9-77-VOL-1-3 NL

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER: CSM UM 9-77, Volume I 2. GOVT ACCESSION NO.: AD-A086160	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle): THE CCTC QUICK-REACTING GENERAL WAR GAMING SYSTEM (QUICK), Users Manual, Data Management Subsystem	5. TYPE OF REPORT & PERIOD COVERED 6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Dale J. Sanders Paul F. M. Maykrantz Jim M. Herrin Edward F. Bersson	8. CONTRACT OR GRANT NUMBER(s): DCA 100-75-C-0019
9. PERFORMING ORGANIZATION NAME AND ADDRESS: System Sciences, Incorporated 4720 Montgomery Lane Bethesda, Maryland 20014	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Command and Control Technical Center Room BE-685, The Pentagon, Washington, DC 20301	12. REPORT DATE: 1 June 1977 13. NUMBER OF PAGES: 170
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report): Approved for public release; distribution unlimited.	
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)	
18. SUPPLEMENTARY NOTES	
19. KEY WORDS (Continue on reverse side if necessary and identify by block number): War Gaming, Resource Allocation	

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The computerized Quick-Reacting General War Gaming System (QUICK) will accept input data, automatically generate global strategic nuclear war plans, provide statistical output summaries, and produce input tapes to simulator subsystems external to QUICK.

The Users Manual consists of four volumes which are intended for the user/analyst who is concerned with preparing the data base for a war game, selecting optional features of QUICK, designating control parameters, submitting computer jobs, and analyzing computer output. This volume, Volume I, provides detailed

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE S/N 0102-014-6601

UNCLASSIFIED

CR-3

20. ABSTRACT (Continued).
instructions for execution of the Data Management Subsystem and the modules it comprises.

The Users Manual complements the other QUICK Manuals to facilitate application of the war gaming system. These manuals (Series 9-77) are published by the Command and Control Technical Center (CCTC), Defense Communications Agency (DCA), The Pentagon, Washington, DC 20301.